Title
Towards Automatic Visual Recognition of Horse Pain

Author
Rashid, Maheen

Publication Date
2021

Peer reviewed|Thesis/dissertation

Towards Automatic Visual Recognition of Horse Pain

By

MAHEEN RASHID

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Yong Jae Lee, Chair

Pia Haubro Andersen

Zhou Yu

Committee in Charge

2021

For my parents

Abstract

Pain is a manifestation of disease and decreases welfare. Early detection of animal pain can not only improve animal well being by enabling early diagnosis and treatment of disease, but can also reduce healthcare costs for livestock owners. A video based animal pain detection system can provide a reliable and scalable means to unobtrusively monitor animals round the clock for signs of pain behavior, and enable the timely provision of medical treatment and pain management. This thesis presents the first steps towards creating such an automated visual system for animal pain detection. In particular, it presents computer vision techniques for pain recognition in the horse, and addresses the challenges of reliably determining pain when working with small scale and sparsely annotated datasets.

We first present two methods that address challenges in veterinary research on equine pain detection by transferring techniques from computer vision and graph theory. We present a unifying description of the equine pain face by use of the biologically grounded language of Equine Facial Action Coding System (EquiFACS) to identify facial changes most correlated with pain in horses. In addition, we develop a novel graph based method that deduces the components of pain expression in horses by inspecting correlations between facial changes. Following, we develop an automatic and easy to use application for finding horse faces in videos that allows veterinary researchers to quickly identify time segments suitable for facial expression annotation from long videos. The application uses a deep convolution network for fast and reliable detection of horse faces, saves veterinary researchers hours in valuable annotation time, enables blinding during the data selection process, and has been

instrumental in the description of both the pain and the stress face in horses in terms of EquiFACS.

Beyond veterinary science, we present novel computer vision methods for automatic horse behavior understanding that use small and sparsely annotated datasets. We present a means for identifying facial keypoints in animal faces that enables accurate detection of horse facial parts without requiring large amounts of training data by transferring knowledge from large, readily available human facial keypoint datasets via face structure warping. Apart from facial keypoints, collecting detailed horse pain annotation in videos is cumbersome, and unscalable. We address this problem by developing two different methods that are capable of identifying pain behavior with crude – weak – video level training labels. First, we present a graph convolution based method for action localization that, by the explicit use of similarity relationships between time segments in videos, temporally localizes the extent of actions in videos despite not being trained with any such annotation. Finally, we present a method for pain detection in horses that uses horse pose cues, learned via multi-view surveillance footage, in a weakly supervised setting to deduce the pain status of the horse. The method identifies pain features that align well with pain scales currently used by veterinary practitioners, with impressive accuracy.

Prof. Yong Jae Lee

Dissertation Committee Chair

Acknowledgements

My time at Davis has been a period of intense professional and personal growth, and it would not have been possible without the army of colleagues, mentors, friends, and family that have supported me at every turn.

First, a big thank you to my PhD advisor, Yong Jae Lee for teaching me how to be a dedicated, informed, and ethical researcher. Yong Jae has taught me to look beyond the immediate problem towards the bigger research questions, be a good citizen of the scientific community, and to focus on quality and rigour in research. He has always encouraged me to work on problems that pique my interest and supported my research ideas. Despite being a brilliant and busy researcher, Yong Jae is humble, earnest, kind and has always made himself available whenever I needed advice or help, and I aspire to be a similarly accommodating, grounded, and dedicated mentor in the future.

Thank you also to Pia Haubro Andersen, the catalyst behind this thesis topic. I did not know anything about horses when I started working on this project, and all that I have learned from the veterinary side is owed to Pia. Pia has been an incredibly reassuring and encouraging mentor, and has taught me the value of passion, vision, and empathy in creating and sustaining long term research projects and relationships. Whenever I was discouraged and found it difficult to take a birds eye view of my research topic and its broader social impact, even a short conversation with Pia would leave me feeling inspired and motivated. In light of COVID riddled 2020, I cannot overstate how necessary these conversations were to get this thesis to the finishing line.

I thank my PhD qualifier committee, Professor Pia Haubro Andersen, Professor Ian Davidson, Professor Ilias Tagkopoulos, and Professor Zhou Yu for their valuable and constructive feedback. Thank you to Professor Yu and Andersen for additionally serving on my PhD dissertation committee. A big thanks to the staff at the UCD Computer Science department, particularly Jessica Stoller, Alyssa Bates, and Jane Ryan for assisting in administrative and financial matters, and making the PhD process smooth.

Throughout the ups and downs of research, my lab mates have consistently made going to work a complete pleasure. Thank you Fanyi Xiao, Krishna Kumar Singh, Chongruo Wu, Jason Ren, Wenjian Hu, Utkarsh Ojha, Yuheng Li, Yangming Wen, Tyler Jan, Markham Anderson, Haotian Liu, and Xueyan Zou for creating a friendly, supportive and collaborative work space, providing fruitful and enlightening discussion on research, and indulging my goofy disruptions. In particular, thank you to Krishna and Fanyi who, apart from being formidable researchers, have provided stimulating and entertaining debate on every topic under the sun since day one of my PhD. I will miss and cherish our Hunan Bar dinners very much.

I have been fortunate to collaborate with talented and dedicated researchers from both machine learning and veterinary science throughout my Phd. A big thank you to Xiuye Gu, Sofia Broomé, Hedvig Kjellström, Marie Rhodin, Elin Hernlund, Johan Lundblad, Katrina Ask, Alina Silventoinen, and Karina Gleerup for sharing their expertise and bringing our research projects to fruition. I am especially thankful to Hedvig and Marie for hosting me as a visiting researcher, and to Sofia for being my Swedish guide and comrade in arms.

This thesis was made possible by a larger collaboration across UC San Diego, UC Davis School of Veterinary Medicine, Swedish University of Agricultural Sciences, and KTH Royal Institute of Technology. In particular, a big thank you to Deborah Forster, and Claudia Sonder whose efforts kick-started the conversations that resulted in this thesis.

I would not have started a PhD had it not been for my past advisors and mentors: my Masters advisor Martial Hebert, and my undergraduate advisor Sohaib Khan, and mentor

Aamer Zaheer. I doubt that I would have pursued a career in research had I not been taught by Nabil Mustafa, and I thank him very much for setting me on the path of intellectual curiosity that has given my brain joy every day since.

A huge thank you to my friends who built me a home away from home by providing love, humor, and memories. Thank you Johanna Heyer, Samuel Heppelmann, Priya Kshirsagar, Cameron Jones, Tad Dallas, Arjun Menon, Victor Hwang, Firas Abu-Sneneh, Farah Hamade, Matthew Nesvet, Megan Long, Nabeel Akhtar, Mariyam Khalid, Aniqa Arif, Eema Masood, Wajiha Saqib, Maryam Akmal, Kamil Ahsan, Ania Kawiecki, Alejandra Cano, Kylie Mosher, Uta Muller, and Gina Verraster. A special thank you to my then boyfriend now husband, Torbjörn Engström, for being my cheerleader, managing me through an embarrassingly large number of meltdowns with endless good humor, and continuing to stick around.

I am blessed with a large and beautiful extended family who would chide me forever if I do not acknowledge them individually for raising and loving me - Asma Khala, Umar Mamun, Tariq Mamun, Phupho, Gogi Phupho, Kamran Phupha, Asif Phupha, Shabnam Mami, Ayeza Mami, and Nana. I owe the last few and crucial weeks of thesis writing to my aunt, Gogi Phupho, and her family who generously housed and fed me as a COVID refugee forced to quarantine away from home. Thank you.

I owe any grit and fighting spirit I have to my infuriating and brilliant elder brother, Imaad Rashid, who probably knows me best. Thank you to my Dado, whose fortitude, discipline, and narrative skills have always inspired me.

This thesis is for my parents. Thank you for for your unconditional love and support through this long journey, for teaching me the value of dedication and honesty, and for providing me with a model of kindness, generosity, and morality to aspire towards. I love you very much.

Contents

Chapter 1

Introduction

Recognition of pain in animals is important because pain is a manifestation of disease and decreases animal welfare. Early recognition of pain can prevent unnecessary animal suffering by allowing for timely provision of pain relief and medical treatment.

Increasing animal welfare can be of particular benefit to livestock practitioners who are under increasing pressure to provide humane living conditions to livestock from conscientious and aware consumers [1]. For example, animal welfare concerns was the biggest reason for people to switch to vegan and vegetarian diets in the UK in the past five years [2]. Early detection of pain can also help livestock owners save in medical and veterinary costs which are significant. For example, an estimated 2.3 billion is spent nationwide on veterinary expenditure for cattle [3, 4], and according to a survey conducted in 2015 by the Center for Equine Health at UC Davis of more than 3000 horse owners, the average veterinary and medical expenditure of a single horse can total $2500 annually.

The International Association for the Study of Pain defines pain as "an unpleasant sensory and emotional experience associated with actual or potential tissue damage or described in terms of such damage" [5]. The emotional component of pain is labelled as 'affective' or 'aversive' in animals [6], and represents awareness by the animal of damage or threat to the integrity of its tissues. As a sensory experience, it changes the animal's physiology and

behavior in order to reduce or avoid perceived damage, prevent its recurrence, and promote recovery. Treating pain as an emotional experience, modern pain scales are primarily based on behavioral parameters rather than physiological measures [7, 8, 9, 10, 11, 12, 13, 14, 15]. Alongside identifying body movements indicative of pain, facial expressions of pain, the 'pain face', has been identified for horses [16, 17], cows [13], and sheep [14] among other mammals. In all three mammals, the pain face is considered a highly reliable indicator for the presence of acute pain.

However, identification of animal pain through human observation is problematic. It can lead to inaccurate evaluations because prey animals, such as horses and cattle, display less obvious pain behavior and hide pain symptoms around unfamiliar human observers [18, 19]. It can also lead to incomplete evaluations because accurate assessment of pain may require constant 24 hour surveillance which is impractical for a human observer.

This thesis envisions a solution to the above problems through an automatic visual animal pain detection system. If pain could be assessed continuously and automatically from video, footage from installed surveillance cameras could be analyzed by artificially intelligent systems to monitor animal behavior round the clock and identify animals that exhibit signs of distress. The animals could then be provided with timely medical treatment. Such a system would help reduce both the distress of monitored animals, and the medical cost of treatment by allowing for early diagnosis.

In particular, this thesis lays the ground work for an automatic pain monitoring system by developing computer vision techniques for pain recognition in the horse, and addresses the challenges of reliably determining pain when working with small scale and sparsely annotated datasets. While the methods presented have been developed for equine pain recognition, my hope is that they will be successfully adapted and extended for other livestock species and provide a broader impact on animal welfare.

Computer vision addresses the technology that enables intelligent processing of visual data. Since the wide scale adoption of deep learning after AlexNet [20], research in com-

puter vision has made great progress for a wide range of problems such as object detection [21], video summarization [22], facial recognition [23], and visual question answering [24]. Computer vision techniques are ubiquitous in every day life; for example, computer vision models help tag, share, and organize photo collections, provide personalized product recommendations and ads, filter content on search engines, and help navigate self driving cars.

Computer vision research has also shown promising results on animal pain detection. Promising results have been shown on sheep [25], mice [26], donkeys [27], and horses [28] and point to the promise of this area of research.

While promising, research on animal pain recognition is nascent, and has not reached the success of computer vision systems on human pain detection. While the seminal Computer Expression Recognition Toolbox [29] that could recognize 6 types of human expressions, used support vector machines, hand crafted Gabor filter features, and small datasets, recent research in human facial expression understanding such as [30, 31] have relied on deep convolution neural networks and large datasets [32, 33, 34] to achieve impressive above 95% accuracy on human facial expression classification.

It is tempting to pursue an approach similar to automatic human pain detection in order to automatically detect equine pain. However, horse pain recognition faces challenges that are unique and distinct from those of human pain detection, and demands tailor made solutions.

Human pain research has been primarily informed by the gold standard of self reported pain. Humans self report their level of pain, and the facial expressions associated with that time have been used to determine what their pain looks like [35]. Facial expressions of pain were described using Facial Action Coding System [36]. The coding system comprises of Action Units (AU) that describe how contraction of specific facial muscle fibres change facial appearance. Since it is grounded in the underlying anatomy of the skin of the face, FACS presents an objective language for describing facial changes, that is agnostic to the experimental setup or method where it is used. FACS based annotation also lead to the

development of large and reliably annotated datasets [33, 37, 38, 39], that have been used to train expression detection systems. As deep learning systems required more data for training, keyword based image searches have been used in combination with automatic methods to build larger datasets [40].

On the other hand, being non-verbal, a gold standard based on self-reported pain does not exist for horses. Consequently many different descriptive languages exist to describe the symptoms of pain. For example, the horse ear positions indicative of pain have been described varyingly as "stiffly backwards" [16], "backwards" [11], "lowered", and "asymmetric" [13]. Lack of self report also makes it difficult to account for temporal variability in pain intensity, and behavioral differences between individuals and breeds in pain expression.

Furthermore, at the start of my PhD, usable horse video or image datasets were non-existent. This was partly due to the nascence of this area of interdisciplinary research. The equine facial expressions of pain described in veterinary studies in 2014 [16] and 2015 [13], while the pain face in humans was described in 1985 [41]. As for humans, annotating equine video data with facial action codes is a cumbersome process, with a single minute video clip taking from 30 to 120 minutes to annotate, depending on the contents of the video.

Additionally, collecting horse video datasets comes with the added overhead of ensuring that footage features the horse in frame, clearly visible, and not occluded. In the study by Gleerup et al [13], weeks were spent to train six horses to stand comfortably and still in front of a camera, which makes such an approach cumbersome, unscalable, and probably provides an unnatural representation of spontaneous pain. In fact, data that has been collected with humans present may fail to generalize well since horses have been shown to hide affective state in the presence of human observers [19]. Alternatively, surveillance footage of horses could be sieved for usable time periods as done by Dalla Costa et al [16]. However, the process of manually selecting usable video segments or frames can take as long as the video footage itself, and hampers the necessary blinding process for data selection.

Interestingly, human detection of equine pain can be far from excellent. Veterinary

experts have been shown to have an accuracy of just 58% in correctly identifying horse pain from videos [28]. These results echo findings in human pain detection, where human health care providers' are show to have low ability to correctly recognize human pain [42].

These challenges have informed the work done in this thesis, and are presented next.

Chapter 2 addresses the need for a unifying description of the equine pain face that is grounded in an objective and method agnostic language. Specifically, the Equine Facial Action Coding System [43] is used to describe the expression of pain in horses using both a novel method developed for this thesis, alongside an established method used previously for human facial expressions. The method of Kunz et al [44], that was previously used to describe the human pain face, is used to deduce the set of equine action units most correlated with pain in horses by inspecting the frequency of action units' occurrence. Additionally, a novel graph based method that uses the correlations between action units to infer the equine expression of pain is presented. Furthermore, we inspect how the expression of pain may vary over different lengths of time, and across clinical and experimental settings. The work also compares and contrasts the features of pain presented across different studies with those presented by us.

In Chapter 3 we address the pragmatic problem of video selection overhead when annotating equine datasets. We present a method that automates the process of finding time sequences in long videos of unobserved horses where the horse face is clearly visible and suitable for further facial expression annotation. The method used a retrained YOLO v2 [45] object detector to detect horse faces in side and front view in frames extracted from an input video. The detections' confidence across video time are then shown to the user for them to prioritize their annotation efforts and improve productivity by going from most usable to least usable video segments. The method has an easy to use graphical user interface, reduced video sequence selection overhead to a percent of what it was before, and has been used extensively for selecting video clips for equine facial action coding in video footage of horses at Swedish University of Agricultural Sciences, most recently in [46].

Chapter 4 presents a method for detecting facial keypoints in animals – specifically horses, and sheep. The method additionally addresses the challenge of animal dataset availability by transferring information from readily available human facial keypoint datasets. Keypoint detection is an important prerequisite for alignment of faces before facial expression analysis, and has been used for animal expression analysis [25, 47, 27]. Given an input image of an animal face, the system outputs the pixel location of the eye centers, mouth corners and nose center of the horse. The method localized facial keypoints on animals by transferring knowledge gained from human faces. A conventional approach to the problem would be to finetune a network trained to detect keypoints on human faces to horse faces. We showed such an approach to be sub-optimal due the large difference in face shape between the two species. Instead we warped animal faces to resemble a more human shape, and then used the warped animal face images to finetune a pre-trained human keypoint detection network. In this way we were able to compensate for the lack of a large animal keypoint dataset by effectively transferring information from human datasets, and additionally directly addressed the problem of dataset availability by building and publicly releasing a dataset of horse facial images and keypoints comprising of more than 3000 images.

Given a video of a horse, it can be cumbersome to annotate the exact start and end times of segments where the horse is expressing pain. Furthermore, due to lack of a gold standard for pain it may be difficult to ascertain which visual features to annotate as painful or not painful. However, it can be much easier to annotate the entire video as a pain video if it contains (or is likely to contain) parts where the horse expresses pain. These 'weak' video level labels can then be used to train a multi-instance learning system that can not only distinguish between videos with pain and videos without pain, but can also indicate the exact times when the expression of pain becomes visible. In computer vision, this problem has been explored through weakly supervised action localization and classification and forms the subject of Chapter 5 and Chapter 6.

In Chapter 5 we present a method for weakly-supervised action localization using a novel

graph convolution based network. In order to find and classify video time segments that correspond to relevant action classes, a system must be able to both identify discriminative time segments in each video, and identify the full extent of each action. Achieving this with weak video level labels requires the system to use similarity and dissimilarity between moments across videos in the training data to understand both how an action appears, as well as the sub-actions that comprise the action's full extent. However, previous methods do not make explicit use of similarity between video moments to inform the localization and classification predictions. Our method used graph convolutions to explicitly model appearance and motion similarity between video moments. Video moments were represented as nodes in a graph, and the edge between any two graph nodes was weighed by how similar two nodes were. By performing inference over this graph during both train and test time, we were able to explicitly model similarity relationships to successfully localize as well as classify actions in video. The proposed method generalized well across three different computer vision action localization benchmark datasets, and can also be applied for horse pain recognition.

Chapter 6 directly addresses the problem of equine pain recognition from surveillance video footage. The dataset features horses in box stalls with induced orthopaedic pain that are filmed using four surveillance cameras. Training a deep network to infer pain directly from the video frames is shown to be prone to overfitting. At the same time, it is not possible to ensure that the network does not focus on extraneous information, such as the filming time stamp or box stall lighting to determine the pain status of the horse. Therefore, we use a self-supervised method to disentangle the horse pose from its identity, and from the background. The disentangled pose representation is then used to determine the pain status of the horse. Furthermore, we propose a new pain specific loss formulation that is able to use weak labels effectively and show that it outperforms both a strong supervision baseline, and the more classic multi-instance learning loss formulation.

The concluding chapter presents a summary of the thesis, and its limitations. In addition, I present an overview of directions in which research in this area is growing, and exciting

directions for future research and development.

1.1 Contributions

This dissertation presents various approaches to address the challenges of equine pain detection. As an interdisciplinary area of research, I present work that is useful for veterinary research on pain (Chapters 1 and 2), as well as novel methods in computer vision that address the technical challenges of equine pain understanding (Chapters 3-5). Following are the contributions of this dissertation:

- Described the equine pain face in terms of Equine Facial Action Coding System. As EquiFACS is grounded in the musculature of the horse face, the language used to describe the pain face is both objective and comprehensive. We found inner brow raiser to be less important than previously believed, and found the half blink and chin raiser to be important indicators of pain.

- Proposed a novel graph based method for determining the facial changes associated with an emotional state by modeling the co-occurrences between facial muscle movements in a pain and no-pain state. Unlike previous methods developed to determine human emotion expressions [44], our Co-Occurrence Graph method is able to model co-occurrences over varying temporal extents, and can highlight facial changes that are infrequent, but discriminative. The method has also been applied to determine the equine face of stress [46]

- Developed the Horse Face Finder, a deep convolution network and software for localizing horse faces in video frames. The method reduces the overhead time for EquiFACS annotation in long videos by 0.01 times, while reducing the likelihood of data selection bias by allowing for blinded data selection. The software has been used extensively at Swedish University of Agricultural Sciences, including data presented in [46] and [48].

- Created a novel method for animal facial keypoint detection based on transfer learning from human keypoint datasets, and deep networks. The method achieved impressive performance on horses and sheep with just 8% and 0.8% failure rates respectively.

- Created and publicly released a horse facial keypoint dataset, comprising of 3171 images of horse faces with eyes, nose, and mouth corners marked.

- Circumvented the need for detailed temporal annotation for pain by proposing a method for action localization supervised with weak classification labels. The method made explicit use of similarity relationships between time segments with the use of graph convolutions and pushed the state of the art in weakly supervised action localization across three benchmark datasets.

- Used self-supervision and temporally aligned multi-view videos of horses to learn a latent representation of the horse pose that is disentangled from the scene background and horse identity.

- Developed a novel multi-instance learning loss for weakly supervised pain detection in horses.

- Used the disentangled pose representation and novel pain loss to develop a method for equine pain detection in unobserved horses. The method achieves up to 65% accuracy in pain vs no pain classification.

Chapter 2

Equine Facial Action Coding System for determination of pain-related facial responses in videos of horses

Pain is a sign of disease, and early recognition of pain may improve welfare and treatment of otherwise disabling diseases in horses. While self-reporting is the gold standard for assessment of pain in verbal humans [49], there are no measures available for the aversive components of pain in non-verbal mammals, including the horse [50]. The IASP definition of pain states that "the inability to communicate verbally does not negate the possibility that an individual is experiencing pain" [5] , referring to adults, neonates, infants, as well as animals unable to communicate. This has brought attention to communication of pain conveyed by non-verbal behaviours, such as bodily behavior, and visible physiological activity such as muscle tremor and facial expressions. During the last decades, a plethora of pain scales based on pain-related bodily behavior has been developed for horses [7, 8, 9, 10, 11, 12]. Research in facial expressions as indicators of pain in horses is a more recent contribution [16, 17]. In one pain study [17], pain was induced in otherwise healthy and trained horses using short-term acute pain induction models, whereas horses in another study [16] experi-

enced postoperative pain from castration. Despite many differences in the conditions and methodology, these two very different studies identified and described facial activity in the same regions of the face, corresponding to moveable facial muscles related to the ears, eyes, nostrils, lips, and chin. However, differences were also present. Dalla Costa et al. [51] later identified a classifier that could estimate the pain status of the animal based on the facial activities coded, confirming that the categories used for scoring were related to the pain state of the horse. Due to differences in both experimental approaches and descriptions of the facial activities observed, a detailed comparison of the facial activities during pain in the two mentioned studies has not been done.

In humans, the Facial Action Coding System (FACS) provides a recognized method for identifying and recording facial expressions based on the visible movement of the underlying facial muscles [36]. The coding requires extensive training, and reliable coding can be expected from certified coders. Recently, Wathan et al. [43], on the basis of FACS methodology, developed the Equine Facial Action Coding System (EquiFACS) for horses. EquiFACS exhaustively describes all observable equine facial behavior in three categories: 17 Action Units (AUs), four Ear Action Descriptors (EADs) and seven Action Descriptors (ADs). FACS coding uses detailed frame-by-frame video observation of facial muscle movement, as well as changes in facial morphology (e.g., the position of the eyebrows, size/shape of the mouth, lips, or eyelids, the appearance of various furrows, creases, bulges of the skin) to determine which AU(s) occurred. Inter-observer agreement is good-to-excellent for spontaneously generated facial behavior in more than 90% of the action units in humans [52] scored by trained and certified FACS readers.

The work by Wathan [43] showed that facial movements can be coded reliably only from video sequences and provide precise information about times of onset and offset of the individual AUs. The FACS systems exhaustively code all facial activity observed, not only what is thought to be pain-related. Any interpretations of the emotional meaning of the observed AUs occur post-coding, as the coding system itself is entirely atheoretical.

Pain-related facial responses in horses have never been described using EquiFACS. While the methodology now exists for the coding of horse facial activity, no methods exist for the interpretation of the results. Research on *human* facial expressions of pain is mature and extensive. Kunz et al [44] presents a systematic review of studies on human facial expressions of pain and describe current approaches for the identification of AUs associated with pain. One approach for defining an AU as pain related is for it to occur frequently, i.e. forming more than 5%, a heuristically set limit, of total AU occurrences in pain state [53]. The second approach is to define an AU as pain related if it occurs more frequently during pain than during baseline [35]. Most often both criteria are applied after each other [54], resulting in a set of AUs that are both frequent and distinct to pain. These methods have never been investigated in horses. Additionally they do not take into consideration the temporal patterns of co-occurring facials actions into consideration, which are increasingly recognized as important for interpretation of facial expressions [55].

Therefore, the aims of this study were to code facial expressions of horses before and during acute experimental pain, and to develop and test statistical approaches that define pain-related facial movements in EquiFACS.

We used videos from a published experiment of acute pain [17] where the horses were habituated to the surroundings and filming conditions, rendering the horses minimally influenced by external input. To explore our models' ability to generalize to horses in a less controlled environment we also collected and EquiFACS coded videos of horses with and without pain in a clinical setting.

We expected that EquiFACS analysis of painful horses would indicate facial activities in the same anatomical regions as pointed out by the Horse Grimace Scale [16] and the Pain Face [17], and that the statistics based on frequency and the temporal information of the EquiFACS coding could be used to identify facial expressions of pain in videos of horses with experimental and spontaneously occurring pain.

2.1 Horse Pain Dataset

2.1.1 Experimental Pain Data

We used videos of six healthy horses of different breeds, five mares and one gelding, aged 3–14 years, recorded during a study of horses subject to acute short-term pain [17]. Briefly, horses were stabled at the research facility for at least ten days before the study, and were positively reinforced during this time to stand in the trial area while wearing only a neck collar. These conditions were designed to increase the horse's comfort in trial settings, reducing the risk of external factors influencing the horse. Baseline recordings (using Canon Legria HF S21, Canon Inc., Tokyo, Japan) were obtained on the day of the experiment. Acute short term ischemic pain was induced by the application of a pneumatic blood pressure cuff placed on a forelimb and the session was recorded for 20 minutes, while pain behaviour was observed and scored using a modified version of a composite measure pain scale [56].

Video clips of 30 seconds duration were selected from the baseline period, and during nociceptive stimulation, at the first occasion where the profiled horse was within the frame for 30 seconds. This resulted in two videos per horse before and during pain, totaling twelve videos.

2.1.2 Clinical Pain Data

Twenty-one horses admitted to a horse clinic for either treatment of a disease (n=11), or control/farriery (n=10) were filmed with a handheld video camera (Canon Legria, Tokyo, Japan, in HD quality), not restrained and in their observation stall in the premises of The University Animal Hospital Copenhagen or Sweden. Their age ranged from 3 to 17 years (median 8 years) and breeds included warm blood horses (n=11), trotters (n=8) and Icelandic horses (n=2). They were filmed from outside the box with hand held cameras, at the earliest 6 hours after being installed in the box without further acclimatization. Inclusion criteria were owners' consent for research purposes and exclusion criteria were horses that displayed

obvious bodily pain behaviour.

Three veterinarians (two females and one male, with more than 10 years of personal experience with horses) assessed pain level based on their clinical experience as either 'Severe Pain', 'Moderate Pain', or 'No Pain', for each horse without prior knowledge of the horses. To obtain a single pain label, we used majority voting between raters. That is, if at least two of the three raters labeled a video as either 'Moderate' or 'Severe' pain, the video was labeled as 'Pain', else the video was labeled as 'No Pain'. This resulted in 7 pain and 14 no-pain videos. The video clips were FACS annotated by a single certified EquiFACS coder without prior knowledge of the horses.

2.1.3 Equine Facial Action Coding System

Equine Facial Action Coding System, as described by Wathan et al. [43] was used for a complete annotation of all videos. The system consists of 17 Action Units (AUs), and 11 Action descriptors (ADs), of which four are Ear Action Descriptors (EADs). While AUs represent the contraction of a particular muscle or muscle group, ADs describe a movement caused by either an undetermined muscular basis, or by deep muscles. *For simplicity all EquiFACS codes are referred to as AUs in the following text.*

All films were coded in a blinded manner by a single certified EquiFACS coder without knowledge of the study horses with inter-rater agreement $> 70\%$ and intra-rater agreement 93%. A complete list of the 28 codes [43] were entered into the annotation software (freeware ELAN [57]). The video clip was first viewed in normal speed. Following, over at least three slow motion, or frame-by-frame, re-runs the annotator coded three regions of the horse face - the ears, upper face, and lower face - and noted the appearance and disappearance of all facial activity. In addition, it was noted if a specific region was out of the frame and therefore not codable.

The resulting dataset contains the occurrence of different AUs, time of their onset, offset, duration, and their temporal overlap with other active AUs. In the statistics section we refer

to each period of AU activation – the contraction of muscle or muscle groups associated with the AU – as an AU occurrence. The duration of an AU occurrence is the period of time that elapses between the start and end of its activation. The frequency of an AU in a video sequence is the number of times it is activated during the video for that AU.

2.2 Discovering Pain AUs

The EquiFACS datasets derived from experimental and clinical videos was used to identify the action units most useful for the identification of pain in a data-driven manner. AUs associated with head and neck movement were excluded as they do not correspond with facial expressions.

We use a paired t-test for mean values for experimental data, and unpaired t-test for mean values for clinical data to test significance. The number of times an AU occurs within an observation was used for the t-test.

2.2.1 Human FACS Interpretation (HFI) Method

As laid out by Kunz et al in a systematic review on human facial expressions of pain [44], we used a two step approach in determining pain AUs. First, AUs that form more than 5% of all AU occurrences in pain videos were selected, meaning that an AU was selected if the number of times it was active in pain videos formed more than 5% of the total number of times any AU was active. From these, the AUs that occurred more frequently in pain than in no-pain videos were determined as the final pain related AUs. To account for unequal number of pain and no-pain videos, AU frequency for pain and no-pain groups was normalized by the number of videos in each group before comparison.

2.2.2 Co-occurrence Method

While the method presented above (Section 2.2.1) was simple, it does not take into consideration the temporal distribution of onset-offset of the various AUs. AUs that comprise a pain expression are likely to co-occur, i.e. occur together, in a pain state, and are likely to co-occur with a different set of AUs in a no-pain state.

We therefore developed a novel method for describing pain expressions by identifying AUs that occur together in a given period of time. Instead of looking at only frequency and distinctiveness, we compared patterns of co-occurrence of AUs between pain and no-pain states to discover the AUs most indicative of pain.

For comparison of the patterns, we built a graph to capture the co-occurrence relationships between AUs. Each node represented an AU and edges between nodes were weighted by how often they occurred together. We then inspected how edge weights changed between pain and no-pain videos, and selected AUs that exhibited the largest change as pain AUs. All AUs that were active during a pre-defined slice of time – an Observation Window – were counted as co-occurring. This information was available since we recorded the start and end time of each AU activation (see Section 2.1.3).

More specifically, we built a 'Co-occurrence Graph' each for pain – G_P – and no-pain – G_{NP} – states. The graph was represented as a $N \times N$ adjacency matrix, where N is the total number of annotated action units, and value in row i and column j of the matrix represents the edge from AU i to AU j, and is weighted by the fraction of times AU j occurs in the same Observation Window as action unit i. For example, if AU j occurs together with AU i in 5 time slices, and AU i occurs in 10 time slices in total the value in row i, column j – referred to as $G^{i,j}$ – would be $5/10 = 0.5$. The diagonal of this matrix was set to zero. The Co-occurrence Graph is directed, meaning that the value in $G^{i,j}$ need not equal $G^{j,i}$ since AU i and j can occur in a different total number of Observation Windows.

Using fraction, or relative co-occurrence, rather than raw co-occurrence count, to weigh each edge acts as a normalization procedure such that AUs that occur more frequently (such

as blinking) do not have higher edge weights than AUs that occur less frequently. Edge values also become easily interpretable as they capture the co-occurrence rate of any two AUs relative to other co-occurring AUs, and are bounded between 0 and 1.

Following, we subtracted the adjacency matrix of no-pain co-occurrence graph from the adjacency matrix of pain co-occurrence graph to obtain a 'Difference Graph', G_D.

$$G_D = G_P - G_{NP}$$

G_D captures changes in relative co-occurrence importance between pain and no-pain states. For example a difference value of +0.3 between AU i and j implies that AU j constitutes 30% more of all co-occurrences in pain than it did in no-pain for AU i, and has increased in relative co-occurrence importance. Note that since the pain and no-pain Co-occurrence Graphs are directed graphs, the Difference Graph is also directed.

The AUs with the largest values in the Difference Graph were considered important for pain detection. Given an AU, we calculated its total change in relative co-occurrence importance for all its co-occurring AUs between pain and no-pain states. AUs that showed more than a chosen threshold t change were chosen as pain AUs.

More formally, let $G_D^{i,j}$ be the value in the ith row and jth column in the difference graph adjacency matrix. The importance of AU i to pain detection, r_i, is then calculated by using the following formula that sums column i:

$$r_i = \sum_{j}^{N} \max\{G_D^{j,i},\ 0\}$$

In other words, r_i sums the change in relative co-occurrence AUs that co-occur with AU i experience. Using column wise summation helps highlight AUs that influence the relative co-occurrence of other AUs. A row wise summation, on the other hand, would disproportionately highlight AUs that only exhibit large changes in relative co-occurrence because they occur once or close to once in the entire dataset. By ignoring decreases, or negative values,

in the summation, we avoided AUs that were negatively correlated with pain.

The threshold for selecting pain AUs, t, is done using the following formula, where R is the set of all r_i, $R = \{r_i, \ldots, r_n\}$:

$$t = \alpha \ (\max \ R - \min \ R) + \min \ R$$

where α is a value between 0 and 1. At $\alpha = 0.5$, the threshold is equal to the mid-range of r_i across all AUs.

This selection method thus selects AUs that exhibit a large change in relative co-occurrence between painful and non-painful states. However, the selected AUs need not occur together in the same time slice.

Conjoined Pain AUs

For AUs to configure a pain expression they should co-occur in the same Observation Window. They should also occur more frequently in pain rather than no-pain states. We refer to these as conjoined AUs.

This equates to finding a cluster of AUs in the Difference Graph that are all connected to each other, and have positive edge weights. We used a standard method in graph theory – the Bron-Kerbosch algorithm [58] – to find sets of AUs that satisfy these two conditions. We considered any two AUs, i and j to be connected with a positive edge weight if both $G_D^{i,j}$ and $G_D^{j,i}$ have a positive value. For every set, we summed its positive edge weights in G_D and selected the set with the highest sum as our final conjoined pain AUs.

2.2.3 Observation Window Size (OWS)

The Observation Window Size determines how close in time two AUs must occur to be considered as co-occurring. For example if two AUs occur within the same 5 second slice, with a $OWS = 5$, they would be counted as co-occurring. With longer OWS, more AUs will

probably co-occur, simply because of the continued facial activities of the horse.

Our datasets comprised of 30 second video clips. We used a sliding window based approach to split each video into shorter clips where the step size is set to half the OWS. For example with $OWS = 5$ a 30 second video will be split into 11 shorter clips of duration 5 seconds, starting at times 0, 2.5, 5, 7.5 and so on, seconds. We explored OWS set to $2, 5, 10, 15, 20$ and 30 seconds.

By exploring OWS of increasing length, we could capture AU co-occurrence dynamics of varied time length. Each of these shorter clips were treated as separate pain or no-pain observations. A smaller OWS helps increase the size of our dataset so that more reliable assertions can be made.

2.2.4 Predictive Values

We inspected the power of specific AUs at reliably predicting pain. If the AU, or set of AUs, are active in a video clip, we marked it as a pain video. Otherwise we marked it as a no pain video. These pain and no-pain predictions were then compared against the ground truth labels to determine the positive and negative predictive value of the AU set.

In addition, we report video level results, where the pain prediction label of the majority observation windows determines the pain prediction label of the entire video.

2.2.5 Pain Observation Probability

Given a randomly selected video segment of fixed time length, we inspect the likelihood of observing AUs found to be associated with pain (pain AUs). We also inspect how this likelihood differs between the pain and no-pain groups.

Specifically, AUs that are associated to pain by both the HFI and Co-Occurrence methods are selected as the pain AUs. For all time segments in the experimental pain dataset of predefined length – the observation window size (OWS) – we report the percentage of time segments that have a given number of pain AUs activated. In addition to the OWS mentioned

Experimental Data Pain AUs with HFI Method

Action Unit	Chin Raiser (AU17)	Nostril Dilator (AD38)	Half Blink (AU47)	Ear Rotator (EAD104)	Eye White Increase (AD1)	Inner Brow Raiser (AU101)	Blink (AU145)	Ears Forward (EAD101)
Percentage of all Pain video AUs	7.23%	10.54%	12.35%	13.86%	5.72%	13.25%	7.83%	8.73%
More Frequent in Pain Videos	✓	✓	✓	✓	✓	✓	✗	✗
Percentage Difference	90.91%	69.23%	56.25%	42.11%	17.14%	2.30%	-14.29%	-18.75%

Table 2.1: AUs found to be associated with pain using the Human FACS Interpretation Method for experimental data.

above (Section 2.2.3), we also used an OWS of 0.04 seconds as a proxy for still image based observation since it corresponds to one frame in a 25 frames per second film. We report the likelihood of observing AUs associated with pain in observation windows from pain videos, as well as no-pain videos. Finally, we inspect the percentage difference in these likelihoods between the pain and no-pain groups. For specified OWS, o, and, number of pain AUs, n, $p_P^{n,o}$ and $p_{NP}^{n,o}$ denote the probability of observing n pain AUs in a time segment of o length in pain videos (P) and no-pain (NP) videos respectively. The percentage difference was then calculated using the following standard formula:

$$\text{Percentage Difference}_{n,t} = \frac{p_P^{n,o} - p_{NP}^{n,o}}{\frac{|p_P^{n,o} + p_{NP}^{n,o}|}{2}} \times 100$$

2.3 Results

2.3.1 Human FACS Interpretation (HFI)

Table 2.1 summarizes the AUs that passed the frequency and distinctiveness criterion for selection, along with the percentage of total AU occurrences each comprised, and the percentage difference in frequency each exhibited between experimental pain and no-pain videos.

Inner brow raiser (AU101), *half blink* (AU47), *chin raiser* (AU17), *ear rotator* (EAD104), *eye white increase* (AD1), and *nostril dilator* (AD38) were associated with pain, while, of the 5% most frequent action units, *blink* (AU145) and *ears forward* (EAD101) were not. Of the selected AUs the most pronounced percentage difference in pain and no-pain frequency is

Experimental Data Pain AUs with Co-Occurrence Method

OWS (sec)	α	Ear Rotator (EAD104)	Half Blink (AU47)	Nostril Dilator (AD38)	Inner Brow Raiser (AU101)	Chin Raiser (AU17)	Eye White Increase (AD1)	Lip Presser (AU24)	Blink (AU145)	Sharp Lip Puller (AU113)	Ears Forward (EAD101)	Chewing (AD81)	Upper Lid Raiser (AU5)	Nostril Lift (AUH13)	Tongue Show (AD19)	Lip Pucker (AU18)
2	0.5	✓	✓	✓	✓	✓	✓					✓				
	0.3	✓ (p<0.01)	✓ (p<0.01)	✓ (p<0.001)	✓ (p=0.720)	✓ (p<0.001)	✓ (p=0.372)	✓ (p=0.660)	✓ (p=0.266)		✓ (p=0.131)	✓ (p<0.001)			✓ (p<0.05)	✓ (p=0.258)
5	0.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	0.3	✓ (p=0.055)	✓ (p<0.01)	✓ (p<0.001)	✓ (p=1.000)	✓ (p<0.001)	✓ (p=0.582)	✓ (p=0.236)	✓ (p=0.695)	✓ (p=0.292)	✓ (p=0.277)	✓ (p<0.01)			✓ (p<0.05)	✓ (p=0.437)
10	0.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
	0.3	✓ (p=0.123)	✓ (p<0.01)	✓ (p<0.001)	✓ (p=0.928)	✓ (p<0.01)	✓ (p=0.712)	✓ (p=0.363)	✓ (p=0.517)	✓ (p=0.344)	✓ (p=0.225)	✓ (p<0.05)	✓ (p=1.000)		✓ (p<0.05)	✓ (p=0.442)
15	0.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			
	0.3	✓ (p=0.160)	✓ (p<0.05)	✓ (p<0.01)	✓ (p=0.740)	✓ (p<0.01)	✓ (p=0.895)	✓ (p=0.415)	✓ (p=0.701)	✓ (p=0.399)	✓ (p=0.240)	✓ (p=0.058)	✓ (p=0.811)		✓ (p=0.066)	✓ (p=0.260)
20	0.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	0.3	✓ (p=0.270)	✓ (p<0.05)	✓ (p<0.05)	✓ (p=0.586)	✓ (p<0.05)	✓ (p=1.000)	✓ (p=0.581)	✓ (p=0.339)	✓ (p=0.504)	✓ (p=0.342)	✓ (p=0.104)	✓ (p=0.662)	✓ (p=0.266)	✓ (p=0.107)	✓ (p=0.389)
30	0.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	0.3	✓ (p=0.175)	✓ (p<0.05)	✓ (p=0.068)	✓ (p=0.920)	✓ (p=0.098)	✓ (p=0.656)	✓ (p=0.733)	✓ (p=0.530)	✓ (p=0.732)	✓ (p=0.621)	✓ (p=0.229)	✓ (p=0.903)	✓ (p=0.296)	✓ (p=0.229)	✓ (p=0.415)

Table 2.2: Pain AUs selected by the Co-Occurrence method for experimental data. Values in parenthesis show p-value using paired t-test for mean values. The threshold values are set to include AUs that are above the mid-range value ($\alpha = 0.5$), as well as above the lower third range value ($\alpha = 0.3$), for change in relative co-occurrence.

for *chin raiser* (AU17) at 90.91%, while *inner brow raiser* (AU101) was barely more frequent in pain videos at just 2.3%.

2.3.2 Co-Occurrence Method

Unlike the HFI Method, the Co-Occurrence method for feature selection relies on temporal information to determine pain AUs. For each OWS we determined the relevant AUs and also reported their p-value. Table 2.2 shows the AUs selected for each observation window size, and for two different threshold values with $\alpha = 0.5$ and 0.3.

Eye white increase (AD1), *chin raiser* (AU17), *nostril dilator* (AD38), *half blink* (AU47), *inner brow raiser* (AU101), and *ear rotator* (EAD104) are selected across all observation window sizes. All of the selected AUs are selected across multiple observation window sizes.

Of the AUs chosen across all OWS, *half blink* (AU47), *nostril dilator* (AD38), and *chin raiser* (AU17) are statistically significant – i.e. with $p < 0.05$ – across almost all OWS. On the other hand, *inner brow raiser* (AU101), and *eye white increase* (AD1) fail to show statistical significance across any observation window size. This is echoed in findings from Section 2.3.1, where *inner brow raiser* (AU101) is barely more frequent in pain videos compared to no-pain videos, and *eye white increase* (AD1) barely constitutes more than 5% of AU occurrences in pain videos.

Using a smaller observation window size not only accounts for briefer periods of pain

Clinical Data Pain AUs with HFI Method

Action Unit	Half Blink (AU47)	Inner Brow Raiser (AU101)	Blink (AU145)	Nostril Dilator (AD38)	Ear Rotator (EAD104)	Ears Forward (EAD101)
Percentage of all Pain video AUs	10.89%	19.76%	17.34%	13.71%	10.89%	9.68%
More Frequent in Pain Videos	✓	✓	✓	✓	✗	✗
Percentage Difference	20.41%	15.38%	7.23%	6.06%	-56.95%	-74.51%

Table 2.3: AUs found to be associated with pain using the Human FACS Interpretation Method for clinical data.

expression, but also increases the number of data points for analysis. As a result with $\alpha = 0.5$, $\sim 71\%$ of AUs selected with an OWS of 2 seconds show statistical significance. In contrast only one, or $\sim 7\%$, of selected AUs show statistical significance when using an observation window size of 30 seconds.

Chewing (AD81), demonstrates statistical significance, and is chosen as a pain AU across almost all OWS. *Chewing* (AD81) is not a frequent action unit, constituting just 2.11% of AU occurrences in pain videos. However, its inclusion demonstrates that it occurs together with other pain AUs and is therefore important.

At $\alpha = 0.3$, more AUs are selected for each OWS, however, the total set of selected AUs across all OWS remains the same.

2.3.3 Conjoined Pain AUs

As described in Section 2.2.2, the conjoined pain AUs occur together in the same time slice, and as a group are more frequent in pain rather than no-pain instances. For brevity, we provide results for $OWS = 2$ seconds. *Nostril dilator* (AD38), *chewing* (AD81), *upper lip raiser* (AU10), *chin raiser* (AU17), and *lip pucker* (AU18) are selected.

2.3.4 Clinical Data

We applied the same methods for deriving pain AUs on the clinical data described in Section 2.1.1. The results using the HFI and Co-Occurrence methods are in Table 2.3 and Ta-

Clinical Data Pain AUs with Co-Occurrence Method

OWS (sec)	α	Nostril Dilator (AD38)	Blink (AU145)	Inner Brow Raiser (AU101)	Nostril Lift (AUH13)	Half Blink (AU47)	Ear Rotator (EAD104)	Ears Forward (EAD101)	Chewing (AD81)	Chin Raiser (AU17)	Jaw Thrust (AD29)	Lip Pucker (AU18)	Lip Presser (AU24)
2	0.5	✓	✓	✓	✓								
	0.3	✓ (p<0.05)	✓ (p=0.473)	✓ (p=0.132)	✓ (p<0.001)		✓ (p<0.001)	✓ (p<0.001)	✓ (p<0.05)				
5	0.5	✓	✓	✓	✓			✓					
	0.3	✓ (p=0.621)	✓ (p=0.904)	✓ (p=0.208)	✓ (p<0.01)		✓ (p<0.001)	✓ (p<0.001)	✓ (p=0.208)				
10	0.5	✓	✓	✓		✓							
	0.3	✓ (p=0.796)	✓ (p=1.000)	✓ (p=0.572)	✓ (p=0.068)	✓ (p=0.491)	✓ (p<0.01)	✓ (p<0.001)	✓ (p=0.373)				
15	0.5	✓	✓	✓		✓	✓						
	0.3	✓ (p=0.725)	✓ (p=0.850)	✓ (p=0.562)	✓ (p=0.227)	✓ (p=0.425)	✓ (p<0.01)	✓ (p<0.001)	✓ (p=0.680)	✓ (p=0.131)			
20	0.5	✓	✓	✓	✓	✓	✓	✓					
	0.3	✓ (p=0.835)	✓ (p=0.853)	✓ (p=0.763)	✓ (p=0.467)	✓ (p=0.463)	✓ (p<0.05)	✓ (p<0.01)	✓ (p=0.775)	✓ (p=0.185)	✓ (p=0.160)	✓ (p<0.05)	
30	0.5	✓	✓	✓	✓	✓	✓	✓					
	0.3	✓ (p=0.889)	✓ (p=0.830)	✓ (p=0.631)	✓ (p=0.580)	✓ (p=0.488)	✓ (p=0.093)	✓ (p<0.05)	✓ (p=0.783)	✓ (p=0.277)	✓ (p=0.163)	✓ (p<0.05)	✓ (p=0.486)

Table 2.4: Pain AUs selected by the Co-Occurrence method for clinical data. Values in parenthesis show p-value using unpaired t-test for mean values. The threshold values are set to include AUs that are above the mid-range value ($\alpha = 0.5$), as well as above the lower third range value ($\alpha = 0.3$), for change in relative co-occurrence.

ble 2.4 respectively. The threshold for co-occurrence AUs was set to the mid-range ($\alpha = 0.5$), and lower third range ($\alpha = 0.3$) as for experimental data.

Conjoined pain AUs for $OWS = 2$ were *jaw thrust* (AD29), *nostril dilator* (AD38), *inner brow raiser* (AU101), and *blink* (AU145).

2.3.5 Specific AUs

As discussed in Section 2.3.1, *inner brow raiser* (AU101) is only slightly more frequent in experimental pain videos than in no-pain videos, with a percentage difference of 2.3%. For the clinical dataset, *inner brow raiser* (AU101) has a much higher percentage difference of 15.38%.

Chin raiser (AU17) and *nostril dilator* (AD38) are selected as AUs indicative of pain by all methods described on experimental data. As a simple test, we use their presence as an indicator of pain and evaluate performance on clinical data.

Table 2.5 (top) shows the positive predictive value (PPV) and negative predictive value (NPV) for pain prediction for each observation. In addition, we report video level results, where the pain prediction of the of majority observation windows determines the pain prediction of the entire video. In either case, the presence of both AU17 and AD38 has a high

Results on Clinical Data Per Observation

	Positive Predictive Value (PPV)				Negative Predictive Value (NPV)			
OWS	AD38	AU17	Either	Both	AD38	AU17	Either	Both
2	38.10%	61.54%	39.11%	85.71%	70.03%	67.92%	71.30%	67.28%
5	35.71%	54.55%	35.97%	77.78%	69.52%	68.90%	70.65%	68.47%
10	33.82%	53.85%	33.33%	83.33%	67.57%	69.57%	66.67%	69.70%
15	32.56%	50.00%	31.25%	80.00%	65.00%	69.81%	60.00%	70.69%
20	32.26%	50.00%	30.30%	66.67%	63.64%	70.59%	55.56%	72.22%
30	31.25%	40.00%	27.78%	66.67%	60.00%	68.75%	33.33%	72.22%

Results on Clinical Data Per Video

	Positive Predictive Value (PPV)				Negative Predictive Value (NPV)			
OWS	AD38	AU17	Either	Both	AD38	AU17	Either	Both
2	37.50%	-	44.44%	-	69.23%	66.67%	75.00%	66.67%
5	30.77%	100.00%	33.33%	-	62.50%	70.00%	66.67%	66.67%
10	35.71%	50.00%	33.33%	100.00%	71.43%	68.42%	66.67%	70.00%
15	33.33%	50.00%	31.25%	100.00%	66.67%	70.59%	60.00%	73.68%
20	31.25%	40.00%	27.78%	66.67%	60.00%	68.75%	33.33%	72.22%
30	31.25%	40.00%	27.78%	66.67%	60.00%	68.75%	33.33%	72.22%

Table 2.5: Positive and negative predictive value for different OWS on clinical data. The criteria for determining pain is the presence of *chin raiser* (AU17), *nostril dilator* (AD38), either, or both. Missing values indicate no observation with required criteria was present.

positive predictive value for all $OWS < 20$. In particular, observing both AUs within the same 15 second interval has an 80% chance of correctly identifying pain. If the majority of 15 second intervals in a 30 second interval show co-occurrence of both AU17 and AD38, then there is a 100% chance of the observation belonging to a pain episode. On the other hand, the absence of both AU17 and AD38 is also a fairly good indicator of no-pain, particularly for $OWS > 5$. Around 7 out of 10 observations where both AUs are absent correctly correspond with no-pain. However around 3 out of 10 times, a pain observation is incorrectly labeled as no-pain.

2.3.6 Probability of Observing Pain

We record the percentage of observations of fixed time length where a given number of AUs associated with pain are found (Section 2.2.5). We use *chin raiser* (AU17), *nostril dilator*

Experimental No Pain Videos

Number of AUs	Observation Window Size (Seconds)						
	0.04	2	5	10	15	20	30
≥1	74.07%	91.95%	98.48%	100.00%	100.00%	100.00%	100.00%
≥2	17.58%	65.52%	84.85%	96.67%	100.00%	100.00%	100.00%
≥3	1.31%	26.44%	62.12%	83.33%	94.44%	91.67%	100.00%
≥4	0.27%	11.49%	30.30%	60.00%	66.67%	75.00%	83.33%
≥5	0.00%	4.02%	15.15%	30.00%	50.00%	58.33%	66.67%
6	0.00%	1.15%	3.03%	10.00%	27.78%	41.67%	50.00%

Experimental Pain Videos

Number of AUs	Observation Window Size (Seconds)						
	0.04	2	5	10	15	20	30
≥1	81.67%	97.70%	100.00%	100.00%	100.00%	100.00%	100.00%
≥2	31.93%	81.03%	96.97%	100.00%	100.00%	100.00%	100.00%
≥3	6.13%	59.20%	84.85%	96.67%	100.00%	100.00%	100.00%
≥4	0.31%	28.16%	72.73%	90.00%	100.00%	100.00%	100.00%
≥5	0.00%	4.02%	24.24%	60.00%	66.67%	66.67%	66.67%
6	0.00%	1.15%	7.58%	23.33%	44.44%	66.67%	66.67%

Table 2.6: Percentage of observation windows from experimental data with specified number of pain AUs present.

(AD38), *half blink* (AU47), *inner brow raiser* (AU101), *eye white increase* (AD1), and *ear rotator* (EAD104) as our pain AUs since they are selected by both the Co-Occurrence, and HFI methods. Results for pain and no-pain videos for experimental data are shown in Table 2.6.

Figure 2.1 shows the percentage difference in probability of observing given number of pain AUs between pain and no-pain videos, i.e. the percentage difference between corresponding cells for pain and no-pain videos in Table 2.6.

The likelihood of observing at least 3 pain AUs is negligible in still frames ($OWS = 0.04$) at $\sim 6\%$, and less than a hundredth chance of observing 4 or more pain AUs. On the other hand, the likelihood of observing at least four pain AUs is much higher for videos, even when observing for 2 seconds at $\sim 28\%$.

The likelihood of observing a range of pain AUs is not negligible in no-pain videos. For example, while 60% of 10 second pain clips display 5 or more pain AUs, 30% of no-pain

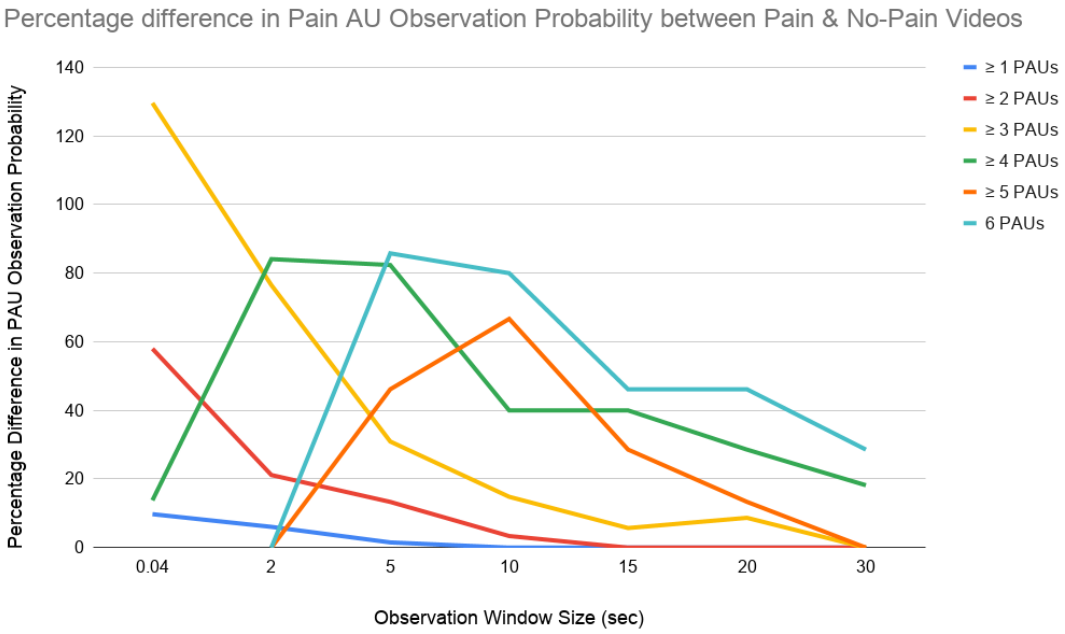

Figure 2.1: Percentage difference between probability of observing given number of Pain AUs (PAUs) in pain videos (Table 2.6 *bottom*) from probability of observing given number of PAUs in no pain videos (Table 2.6 *top*) on experimental data.

10 second clips also display 5 or more pain AUs. As observation window size increases, more AUs can be observed together. At the same time, the difference in AU observation probability is reduced between pain and no-pain videos, with a percentage difference of less than 50% across all AU numbers for OWS greater than or equal to 15 seconds.

2.4 Discussion

This study describes for the first time the facial activities in videos of horses in pain by use of the Equine Facial Action Coding System (EquiFACS) [43]. We explored different statistical methods for the analysis of the EquiFACS data.

Using the HFI method on the experimental data, the two most prevalent AUs in painful horse were the *chin raiser* (AU17) and *nostril dilator* (AD38) (Table 2.1). These two AUs seem to have equivalents in the Horse Grimace Scale [16] as the configurations "mouth strained and pronounced chin" and "strained nostrils and flattening of the profile"; in the

Equine Utrecht University Scale for Facial Assessment of Pain (EQUUS-FAP) scale [11] as the configuration (regarding nostrils) "A bit more opened" or "Obviously more opened, nostril flaring" and "Corners mouth/ Lifted a bit" or "Obviously lifted"; and in the Pain Face [17] as the configuration "Edged shape of the muzzle with lips pressed together" and "Nostril dilated in the medio-lateral direction". This shows that facial expressions of pain as described by EquiFACS occur in the same anatomical regions as described in previous descriptions, such as Pain Face and Horse Grimace Scale.

The third most prevalent AU of the painful horse face was the *half blink* (AU47), which is defined as a reduction of the eye opening by the eyelids, but without complete closure of the eye [43]. The increased rate of half blinks has – to our knowledge – not been documented before as an indication of pain, probably because it is only possible to appreciate this activity from close inspection of video. The action takes place in less than half a second [43]. Decreased eye blink rate has recently been described as a non-invasive measure of stress in horses [59] and the ethogram of the Pain Face contains evidence of increased blinking during pain [17]. The Horse Grimace Scale [16] contains "Orbital Tightening" as a feature with the following description: "The eyelid is partially or completely closed". The description does not specify the duration of the closure of the eyelid, and may correspond to any of *eye closure* (AU143), *half blink* (AU47), or *blink* (AU145). Since EquiFACS uses temporal information during annotation, the type of eye closure can be determined unambiguously.

EQUUS-FAP scale also focuses on the activity in the eye region, but uses both eye closure and eye widening as indicators of pain [11]. The opening of the eye is described as "obviously more opened eyes", and increased visibility of the sclera. In EquiFACS these features would be coded as two separate action units, *upper lid raiser* (AU5), and *eye white increase* (AD1), of which AD1 was found by us to be associated to pain. In other studies, increased visibility of eye white has been associated to stress in horses [60].

The "triangular eye" or "worry wrinkles" has empirically been associated to both stress and pain by horse community peoples and veterinarians [59, 60]. In EquiFACS this appear-

ance is coded as the *inner brow raiser* (AU101). Per definition, the activation of this AU increases the perceived size of the eye region, but not the aperture of the eye [43]. This activity also has a parallel in the Horse Grimace Scale where it is described as "tension above the eye area" [16], and in the Equine Pain Face [13] where it is described as "contraction of m. levator anguli oculi medialis". Given this concurrence we found it remarkable that the frequency of *inner brow raiser* (AU101) was only barely higher in the pain group of this study.

The ears are highly communicative in horses [61]. In this study, increased frequency of *ear rotator* (EAD104) was associated with pain. In the Horse Grimace Scale a "moderately present – stiffly backwards ear" resembles *ear rotator* (EAD104), while the "obviously present stiffly backwards ear" with a wider distance between the tips of the ears resembles the *ear flattener* (EAD103), which has another muscular basis [16]. In the description of the Pain Face, "the lowered ears" with a broader base resembles the *ear rotator* (EAD104), while the "asymmetric ears" described in the Pain Face have no single equivalent in EquiFACS [13]. The EQUUS-FAP scale uses the "backwards ears"; it is not clear if *ear rotator* (EAD104) or *ear flattener* (EAD103) are parallels, or both [11]. It therefore seems important for pain recognition to discriminate between the *ear rotator* (EAD104) and the *ear flattener* (EAD103).

Thus, the EquiFACS and the HFI frequency methods applied from human research point out a number of facial action units that largely correspond well to facial configurations already described in other pain studies. One important exception is the increased frequency of the *half blink* (AU47), which to our knowledge, has not been documented as an action unit with increased frequency during pain. Notably, "the inner brow raiser" (AU101) and the "ears flattener" (EAD103) did not appear as very discriminative of pain.

The HFI method uses each AU frequency independently to determine the subset most correlated with pain. As a result, the selected AUs may not occur at the same time in a pain state. On the other hand, the co-occurrence method captures the relational dynamics of AU

occurrences in observation windows of varying time lengths. As a result, the Co-Occurrence method selects AUs that are likely to be observed at the same time during a pain state and therefore shows the appearance of facial expressions of pain. When the Co-occurrence method was used (Table 2.2) more pain AUs were selected, compared to the HFI method. Generally, AUs of the lower face were selected, specifically *lip pucker* (AU18), *tongue show* (AD19), *lip presser* (AU24), *sharp lip puller* (AU113), and *chewing* (AD81). Regarding nostril movement, *nostril lift* (AUH13) was selected in addition to *nostril dilator* (AD38). Additionally, *eye white increase* (AD1), and *inner brow raiser* (AU101), were selected across all observation time lengths, but were not statistically significant.

While the co-occurrence method identifies AUs that demonstrate a different relational dynamic between pain and no-pain states, the "Conjoined Pain AUs" explicitly identify clusters of AUs that occur together and more frequently in pain than in the no pain states. The method did select both the AUs that demonstrated the strongest association to pain using the HFI and Co-Occurrence methods – *nostril dilator* (AD38), and *chin raiser* (AU17), but also selected AUs associated with lower face movement – *lip pucker* (AU18), *chewing* (AD81) – and nostril movement – *upper lip raiser* (AU10). This may indicate that lower face movements convey indicators of pain that should be further studied.

Not surprisingly, the likelihood of observing multiple pain AUs was strongly linked to the length of observation time. In still images, or OWS of 0.04 seconds, the likelihood of observing more than three pain AUs was negligible at less than half a percent for pain videos, and with little percentage difference from the likelihood of observing the same number of AUs in no-pain videos. In contrast to this, in our limited dataset, observing 4 or more pain AUs in a 5 second observation window was both likely (occurring in 72% of 5 second pain clips), and significantly more likely in a pain video than a no-pain video (percentage difference of 84%). An implication of this may be that observation of video for pain assessment in horses may be of higher value than randomly selected images.

While the experimental dataset was collected under controlled circumstances, with the

pain induction providing a kind of gold standard for the occurrence of pain, no gold standard exists for spontaneous pain. The facial expressions of pain are believed to be universal for all species, across different types of pain [35]. It was therefore of interest to investigate how the models developed from experimental data could predict what clinicians consider to be pain.

For the clinical data set, we deliberately did not infer anything about the diagnoses of horses, since even horses that come for control or routine farriery, may be in pain, and some horses may have diseases that are actually not painful. The true pain status of the horses could not be known, and we can therefore only show how a global pain assessment of clinical cases relates to statistical models built on EquiFACS of experimental horses.

The pain AUs selected by the HFI method were not entirely similar between the clinical and experimental data. While *half blink* (AU47), *nostril dilator* (AD38), and *inner brow raiser* (AU101) were selected as in the experimental data, the AUs *ear rotator* (EAD104), *chin raiser* (AU17), and *eye white increase* (AD1) were not selected. On the other hand, *blink* (AU145), was selected in the clinical data, but was not in the experimental data.

The co-occurrence method selected less AUs in clinical data compared to the experimental data when the threshold for AU selection was similar to the experimental situation. Lowering the selection threshold resulted in a similar set of AUs being selected compared to the experimental data with some exceptions; *Eye white increase* (AD1), *upper lid raiser* (AU5), *sharp lip puller* (AU113), and *tongue show* (AD19) were not selected with clinical data, but were selected with the experimental dataset. On the other hand, *jaw thrust* (AD29) was selected with clinical data, but was not selected with the experimental pain dataset.

Similar to the 5% threshold used in the HFI method, the threshold value α used in the co-occurrence method is set heuristically, and may lead to different results across different datasets. Its value corresponds to the amount of difference AUs must display in co-occurrence patterns between pain and no-pain states to be selected as pain AUs. Developing a criteria for selecting an optimum selection threshold is an important and interesting direction of

future research.

Interesting differences appeared between the clinical and experimental data. AUs corresponding to eye aperture increase (AD1 and AU5) were considered indicative of pain in the experimental dataset, but not in the clinical dataset. Lower face AUs also differed. While experimental data featured *sharp lip puller* (AU113), and *tongue show* (AD19), the clinical data did not and instead featured the *jaw thrust* (AD29). In general, apart from *chewing* (AD81), lower face movements were selected across fewer observation window sizes for clinical data than upper face and nostril movements. We can only speculate about the reasons for these discrepancies, which could be due to differences in the pain experience, pain type (nociceptive acute pain versus chronic or inflammatory pain), pain duration, or reliability of pain/no-pain labels between experimental and clinical data.

The co-occurrence method generally showed overall higher agreement between pain AUs across both datasets than the HFI method. This points to the advantage of co-occurrence over the simple frequency based HFI method. Since the HFI method ignores the temporal dynamics between AUs the method is less able to select discriminative AUs that occur less frequently such as *chin raiser* (AU17). The lack of a gold standard for clinical pain continues to be an unsolved issue. With data that has imperfect labels, the difference between pain and no-pain frequency patterns may be reduced, leading to less consistent results.

To test the pain predictive ability of AUs derived from experimental data in the clinical setting, we used the two AUs most consistently chosen as indicative for pain in the experimental data. The positive predictive values of *nostril dilator* (AU38) and *chin raiser* (AU17) were 100% if these actions were both observed within an Observation Window Size of 10 to 15 seconds. The absence of these actions had a poor negative predictive value, meaning that other actions should be looked for if a horse should be claimed without pain. These observations should be explored further using EquiFACS to increase sensitivity and specificity of pain assessment scales.

One limitation of this pilot study is the low number of experimental horses that the

models were built on. While the acclimatization of horses in the experimental setting was an advantage for obtaining as little interference from external inputs as possible, it might at the same time limit generalisation to data with external interference, where there is no gold standard for assessment of pain. We based the presumption of pain on clinically experienced observers' evaluation, and not the reason for admittance, as the true pain status of these horses can not be known. We used a simple dichotomous pain/no-pain model for this study due to the low number of horses, the lack of a validated pain scale with intensity scoring for video, and the lack of intensity codes in EquiFACS. We could have used both a larger number of experienced clinicians and a larger number of clinical and experimental cases, issues that needed to be balanced against the very resource demanding process of FACS annotation. Finally, this study only investigated the facial activities produced by a single pain modality from experimental data. Clinical data showed more diversity of AUs, which may be due to difficulties with correct pain classification or the co-existence other emotional states. Pain expressions should therefore be studied in a larger number of more diverse horses, during different clinical conditions and with different types of pain.

In conclusion, we have for the first time described the facial activities of one "prototypical" pain face of acute pain in the horse using a Facial Action Coding System. We identified increased frequency of *half blink* (AU47) as an indicator of pain in the horses of this study. The *ear rotator* (EAD104), *nostril dilator* (AD38) and lower face behaviours, particularly *chin raiser* (AU17), were found to be important pain indicators. The *inner brow raiser* (AU101), and *eye white increase* (AD1) had less consistent results across experimental and clinical data. Frequency statistics identified AUs, EADs and ADs that corresponded well to anatomical regions and facial expressions identified by previous horse pain research. Novel co-occurrence based method additionally identified facial behaviors that were pain specific, but not frequent, and showed better generalization between experimental and clinical data. In particular, *chewing* (AD81) was found to be indicative of pain. However, the reported methodologies need further testing in larger sample sizes.

Chapter 3

Horse Face Finder: An automatic tool for assisting EquiFACS annotation

In the previous chapter, we described the pain face in terms of Equine FACS. The next step to using this knowledge in automated equine facial pain detection is the annotation of video data with the relevant pain action units. This is a very time consuming process, as a video sequence must be rewatched in slow motion for each action unit to be annotated. In this chapter we address one problem that additionally makes annotation of horse videos extremely time consuming: finding video segments that are suitable for EquiFACS annotation in the first place. Specifically, we propose a method to automatically find video sequences in long film that feature the horse in a pose with ideal visibility for annotation.

Facial Action Coding System (FACS) is a system to define and name facial movements by their appearance on the face. Originally developed for humans, various mammal specific FACS have been developed [62, 63, 64], including EquiFACS [43] for describing horse facial action movements. Facial action coding systems provides a method for a unique identifying and recording of facial activity, based on the movement of the facial muscles. The system comprises a number of action units, where each action unit describes a specific facial movement produced by the contraction of underlying facial muscle. As a result FACS provides a

comprehensive and objective methodology for annotating and describing facial expressions. In humans, FACS annotated images and videos can be used to identify various emotional states such as happiness, sadness, anger, disgust, as well as pain [65].

FACS for horses, EquiFACS, [43] allows researchers to describe the facial movements of horses in a similarly comprehensive and non-subjective manner. EquiFACS based descriptors therefore hold the potential for description of facial activity related to different internal states of the horse, such as pain, but also stress and fear, stress, fear. If computer vision based systems can used to recognize the relevant facial action units, and ground truth of the internal state can be provided, computer vision based methods maybe developed that can be used in the future research of animals emotional experiences where ground truth generally is difficult to obtain.

The first step towards developing computer based descriptors of emotional states in the horse is the collection and EquiFACS annotation of data. Like human FACS, EquiFACS annotation is a slow and cumbersome process. Observers should be trained and only raters with inter-rater agreements higher than 70% should be used. It can take an EquiFACS expert between 30 to 60 minutes to annotate a single one minute video clip, depending on the number of action units and action descriptors present in the clip. In addition, while it is relatively easy to obtain many sequences suitable to FACS annotation in a human video dataset - humans can be asked to face the camera during filming - this is not the case for freely moving horses. Training horses to hold the same position in front of a camera as done in [17] is not only a time consuming process, but also not possible outside of a clinical study setting. It will also influence the behaviour of the horse. It is, for example, known that horses, being prey animals, can hide pain or other expressions when in the presence of humans [66, 67]. In order to understand spontaneous horse facial activity, it is therefore important to collect data of *unrestrained* and *unobserved* horses.

Filming horses for many hours with surveillance cameras in stalls provides one such setting. With such video footage, EquiFACS annotators would typically first watch a video

at four or twice its speed to initially note time points in the video where the horse face is in side to 45^o angle relative to the camera. They may then then go over the video again at original speed to determine the exact length of the previously identified time points and the quality of video where the horse face is consistently visible. Last, the identified time segments will be annotated. With a predefined time segment length, annotators will spend at least the time of the video to determine annotatable time segments. It is often necessary to identify all annotateable sequences, in order to calculate the sample size.

The aim of our tool is to eliminate time spent in determining video time segments that are suitable for annotation. In addition, the tool allows annotators and researchers to select time segments *randomly* and *blindly*. As a result, our software can be very useful for eliminating different types of bias, for example selection bias (the annotator selects clips e.g. out of convenience) or expectaion bias (the annotator selects clips that fit the expectation about the outcome of the study).

3.1 Related Work

Integrating computer vision and machine learning with veterinary science is a relatively unexplored interdisciplinary area of research. Related work falls in to three broad categories: veterinary research on facial expressions of horses, computer vision research on human facial detection and alignment, and a small but growing body of research on computer vision methods for understanding animal expressions and movement.

Veterinary research on horse facial activity: Facial expressions of pain have been described in the horse in veterinary research in varied settings. In [17], pain was induced in healthy horses and resulting changes in the facial expression were described. In a clinical setting [16] studied horses with post-operative pain from castration where influence of other types of external and internal stimuli could be present. While the above methods studied the undisturbed horses in their boxes, [68], studies the facial expressions of pain in the ridden or

moving horse recently was performed using ethogram developed specifically to that setting. Finally, [11] develop a pain scale using facial movements alongside head movement and gross behaviors on horses with colic and head pain. None of the aboive mentioned studies have used the objective descriptors of facial exxpressions, EquiFACS [43] which provides a system to taxonomize horse facial movements. With 16 action units and 11 action descriptors, EquiFACS can be used to code horse facial activity in any setting, and can consequently be used to study facial expressions in emotional states, such as pain, stress and sedation.

Computer vision for human faces: Human face detection and alignment has a very rich history in computer vision and machine learning research. Face detection methods localize the spatial position and extent of all visible faces in an image. Facial alignment, or key point detection, additionally localize the position of specific points on the face, such as the eye centers, lips corners, nose tip, etc. Many large datasets are available for training and testing face detection systems. Popular datasets include FDDB [69] and WIDER [70] which provide challenging settings for face detection with varying face pose, occlusion, and size. Similarly large datasets are available for human keypoint detection [71, 72]. While the seminal Viola-Jones framework [73] detected faces using a Haar features with Adaboost learning, modern methods rely on deep convolutional networks to achieve both face detection and face alignment, with the state of the art achieving up to 99% average precision on face detection [74], and an error rate of just $\sim 4.04\%$ on face alignment [75].

Computer vision for animal faces: While computer vision methods related to the understanding of human face and expressions are well developed, similar methods are not yet present for horses. Horse detection has been explored indirectly as a subset of objects in widely used object detection datasets such as COCO [76] and PASCAL [77]. Animal facial keypoint detection has been explored through use of shallow [78] and deep [79] learning models, with the latter focusing specifically on horses. Recent work has explored 3D understanding of animal bodies [80, 81]. Automatic detection of pain in images of sheep has

Figure 3.1: **Horse Face Finder:** Video frames are sampled at a user defined frequency, and horse side and front view heads are detected per frame. Results are displayed for users to efficiently determine time segments fit for annotation.

shown promise through use of a shallow model [25], while a deep recurrent model trained and tested on videos of restrained horses has also detected pain accurately[28].

3.2 Approach

Side to 45^o angles are ideal for EquiFACS annotation, since these angles provide a clear view of musculature of the lateral horse face and ears. The Horse Face Finder should automatically find and determine the facial pose of a horse face for time points across a video. Users should be able to see the detected horse face pose information in order to finally determine time segments that are usable. Figure 3.1 gives an overview of our method.

Given an input video, our method first extracts frames from the video at a user defined rate. Every extracted frame is then passed to horse face finding deep convolutional network. The network detects the horse face in side and front view and outputs a confidence value associated with each detection - with 0 indicating no confidence, and 1 indicating maximum confidence in a detection. These detection confidence values are collated and then plotted against video time. The user may then use the plot to prioritize their annotation efforts - starting from time segments with longer and more confident face detections.

3.2.1 The Face Detector

The back bone of our tool is YOLO v2 [45]. YOLO is a deep convolutional neural network based real time object detection system – it is capable of determining the spatial position and extent of predefined visual categories (such as car, dog, horse etc) in input images. We adapt YOLO to detect two types of 'objects' - horse faces in side view, and horse faces in front view. Even though face side views are most useful for EquiFACS, front views can be important for assessing symmetry, and some ear positions. At the same time, training the network with these two categories rather than just a side view class helps it better distinguish a horse face – regardless of pose – from background.

Training the detector to distinguish and identify these two types of horse faces correctly requires annotated training data: images of horses with front and side view faces marked. We used the dataset from [79] that had horse face bounding box annotation and manually removed images that did not have the full horse face visible. We further marked each annotated horse face as either side view, front view, or neither if the face was mostly self-occluded. In addition we manually annotated and added frames from two twenty minute surveillance videos of horses. The addition of these frames was important to correct the domain difference between the dataset from [79], which comprised of images collected from the internet, and surveillance footage used by our collaborators for EquiFACS annotation. Our final dataset comprised of 3570 training images (of which 524 are from surveillance films), and 177 test images (of which 20 are from surveillance film).

We used the publicly available YOLO implementation to train our system. We trained the model by finetuning the model pretrained on PASCAL [77] for 2000 iterations with a learning rate of 0.0001 and a batch size of 64. Overall, the network can detect horse heads effectively and is unlikely to miss usable time segments. Without images from surveillance videos, our method achieves 87.01% recall. The addition of frames from surveillance videos was important and improved performance by $\sim 8\%$ to 94.92% recall.

3.2.2 Plotting Detections

For every image, our YOLO detector outputs bounding box detections, and confidence values associated with each detection. The detector can predict boxes of two types or 'classes' - face side view boxes and face front view boxes. We use a threshold of 0.2 or 20% confidence to threshold detections - all detections below 0.2 confidence threshold are discarded and the detections with maximum confidence per class are recorded.

The detection confidence are smoothed by averaging over minute long intervals. This helps increase the confidence of video frames with missed detections that are occurring in time periods with high horse face visibility, and can decrease the confidence for time points when the horse face is visible only momentarily.

Both smooth and raw detection confidence values are then plotted against video time. The plot allows the users to know at a glance what time intervals are most suitable for annotation, and how long these time periods last.

3.2.3 Usability

The users of our tool are EquiFACS annotators and veterinarians who may not be familiar with the `Python` backend of the tool. In order to make our tool user friendly, we developed an easy to use Graphical User Interface for our tool. Users can use the tool to select a video, run the face detector, and analyze the resulting detections.

Users are shown a clickable plot of detection confidences against video time. They can select any time period in the video and are show the closest processed video frame and detections. This allows users to very quickly determine parts of the video that feature the horse in a pose ideal for EquiFACS annotation. At the same time, it provides a way for the users to identify and ignore incorrect or noisy detections.

Figure 3.2 shows windows from our simple and easy to use GUI.

3.2.4 Keypoint Detection

Our method is also integrated with previous work [79] that detects facial keypoints in horses. Given a frame with a valid horse head detection – the model from [79] is used to extract the pixel position of the eyes, mouth corners and nose on the detected horse head. The enlarged parts corresponding to each keypoint can then be shown to the user.

Keypoint detection provides an additional way for annotators to asses the suitability of a time segment: the visibility of the facial parts can be evaluated and users can estimate the facial movements the horse is likely to be making at that time. The accuracy of the detected time segments can be used as a proxy for suitability for downstream tasks – such as training a machine learning system for EquiFACS detection. By observing the zoomed in areas around keypoints, users can find or avoid time segments where a specific facial activity, eg. eating, is present. Figure 3.3 (left) shows examples of automatic keypoint detection on some frames in three videos.

3.2.5 Automatic Selection of Time Segments

While the basic tool can be useful for analyzing a single video and determining the best time segments for annotation – manual inspection of very long videos, or many videos can be time consuming. In such cases it can be more useful to provide the user with a list of time segments that are suitable for annotation automatically.

Our tool can provide this additional functionality. Given a user provided desirable time segment length – say 30 seconds – it will automatically find and list time segments of length 30 seconds where the horse face is detected consistently in every frame extracted in that time window. The time segments will further be sorted in decreasing average confidence value. With large volumes of videos, annotators can quickly determine the videos that are most useful. Figure 3.3 (right) shows an example, where surveillance video with the horse not present is given low priority, and a video with the horse standing still and close to the camera is given higher priority.

3.3 Results

We present quantitative and qualitative analysis of our method.

We first present a quantitative analysis of running time on a randomly sampled subset of videos. The larger dataset comprises of 3025 videos, of which 2489 are surveillance videos filming horses in a stall, and 536 show horses being trotted for lameness identification in both indoor and outdoor arenas. From the larger dataset we randomly sample a subset of 170 videos for analysis on running times, and ensure that video length is at least 5 minutes. The videos have a total duration of 152 hours.

Our method comprises of three steps - extracting frames from videos, running the face finder network on extracted frames, and collating and plotting the results. For every video we record the time taken to perform each step in the detection process. Wall clock time is used, and all videos are processed with the help of a Titan-XP GPU. For parallel frame extraction 12 threads are used.

The time taken for each step is divided by the video length to get seconds spent per video minute, and shown in Figure 3.4. Average time taken per video minute across all videos is shown in Table 3.5. By averaging seconds taken per video minute across all tested videos we are able to account for overhead time that is independent of video length.

The most time consuming step in our approach is frame extraction. However, it is possible to eliminate this overhead by running the detector on video input directly which is a future direction of this work. The actual horse head detection takes less than a third of overall running time, taking only an average of 0.21 seconds per video minute. Our method is less efficient for shorter videos as overhead processing time accounts for a larger proportion of total processing time. Overall, processing all 170 videos end to end took 1 hour 45 minutes. The same task would take a human annotator upwards of 38 hours (quarter of total video time).

As detailed in Section 3.2.5, our tool can collate results and save lists of usable time segments per video, as well as across a large dataset of videos. Assuming it takes a human

one minute to scan a single detection plot and record usable segments with the same criteria, a human annotator would take close to 3 hours to perform this listing task. In contrast our tool is able to perform this task for all videos in 0.15 seconds.

In Figure 3.2 we show an overview of the GUI accompanying our tool. The interface is simple and allows annotators to easily asses the quality of face detections, as well as the usability of time segments in a video.

Figure 3.6 shows results from a few videos in our datasets. The detections correspond to time segments when the horse face is visible in side or front views. Apart from lighting and back ground setting, the videos also feature a different gross positioning of the horse relative to the camera, with the camera positioned at the same level as the horse head (first and second row) as well as above the horse head (third row). The tool is able to adapt to these visually different settings easily.

In Figure 3.3 we show the results of keypoint detection on frames from 3 different videos. While the keypoint detection is accurate for frames where the horse head is clearly visible, the results are not correct when the horse is turned away from the camera (last column first and second row), or is occluded (third row). By visually verifying the keypoint detections, annotators can quickly determine the usability of a video time segment not just for EquiFACS annotation, but for downstream tasks such as automatic AU or expression detection.

3.4 Weaknesses

While accurate, the detection of horse head is an imperfect measure of usability of a of a given sequence. In other words, the visibility of a horse head in a desirable pose does not guarantee that it would be possible and desirable to annotate the corresponding time segment. For example, the horse head may only be partially visible, or the horse may be eating, sleeping, or engaged in other activities that prohibit informative EquiFACS annotation. This weakness is due in part to the design of our tool which is agnostic to downstream EquiFACS annotation

protocols. Its effect can be reduced by visual verification of keypoint detection results and future work may choose to modify or prune horse head detections based on task specific requirements.

As pointed out in Section 3.2.1, the face detector is sensitive to visual domains. When testing data dramatically visually different from training data, the face detector will not perform well, and the suggested time segments for annotation will be inaccurate. However, by training the face detector again with additional data from the testing data domain, we can expect such failure cases to decrease. Last, the face detector acts on video frames individually. Information from temporally neighboring frames is not used to make the final prediction. This can lead to missed and false detections that may be avoided if information from neighboring frames were taken in to account. More recent computer vision works deal with the problem of video object detection such as [82], and can be used as the backbone of our tool for better detection.

3.5 Discussion

We present a method for eliminating overhead time in selection of video segments for EquiFACS annotation. The tool determines usable time segments by finding time points where the horse face is visible and in a pose ideal for annotation.

The Horse Face Finder takes as input a video, and detects horse faces in side and front pose. It then outputs a continuous confidence value between 0 and 1 for each time step in the video; 0 indicates that the time step is not usable at all and 1 indicates high confidence in its usability. This confidence value allows users to prioritize their annotation efforts and improve productivity by going from most usable to least usable video segments. In addition, it finds and localizes the horse head in each frame and can automatically list annotatable time segments of user provided duration.

Compared to annotation overhead time of a day with manual inspection, the tool reduces

overhead time to less than 19 minutes on a 24 hour video, with an average running time of 0.77 seconds per video minute. Furthermore, it has a high recall rate of 94.92%.

While developed for EquiFACS annotation, with the availability of training data, it can easily be extended to any animal species. In addition, it can also be extended to other anatomical parts of the body, such as the back of a cow. It therefore has great potential across veterinary science as well as the related research fields of animal science and behavior.

One unique advantage of our tool is that it allows annotators and researchers to select annotatable video segments of pre-defined duration *without* inspecting the actual video footage. As a result proper blinding and randomization will require much less time. Development of methods to quantify and possibly understand the signals given by facial activity understanding is a problematic task and may suffer from different types of selection or expectation bias, as seen in other veterinary disciplines, for example lameness examination [83].

The decoding of facial expressions of pain in animals have found interest in recent years [17, 13, 64] and it is currently accepted that facial expressions may by an under-utilized tool for the assessment of welfare in animals [84].

Previously, from an initial description of the human pain face, research in human pain has expanded to include descriptors of pain in neonates [85], identification of different types of pain faces [86], and machine learning models capable of accurately distinguishing pain from other expressions [29] including faked pain [87].

These advances have taken place on the back of large datasets that were human annotated. For similar success in horse pain or facial activity understanding, it is necessary to collect and annotate datasets of similar scale. Our tool presents a step in this direction by enabling researchers to collect large datasets in an efficient and unbiased manner.

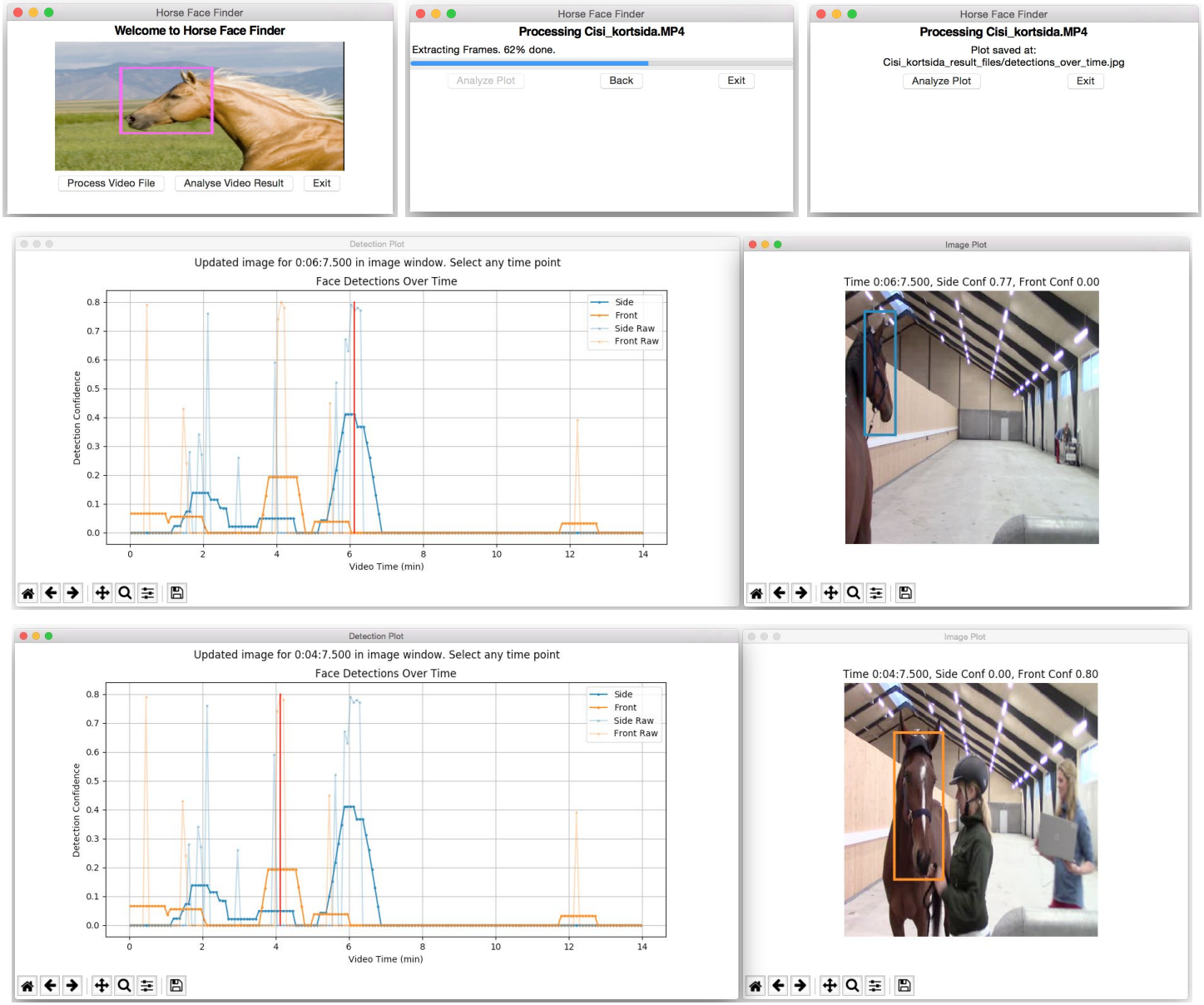

Figure 3.2: The GUI lets users process videos (top row), and analyze results (middle and bottom row).

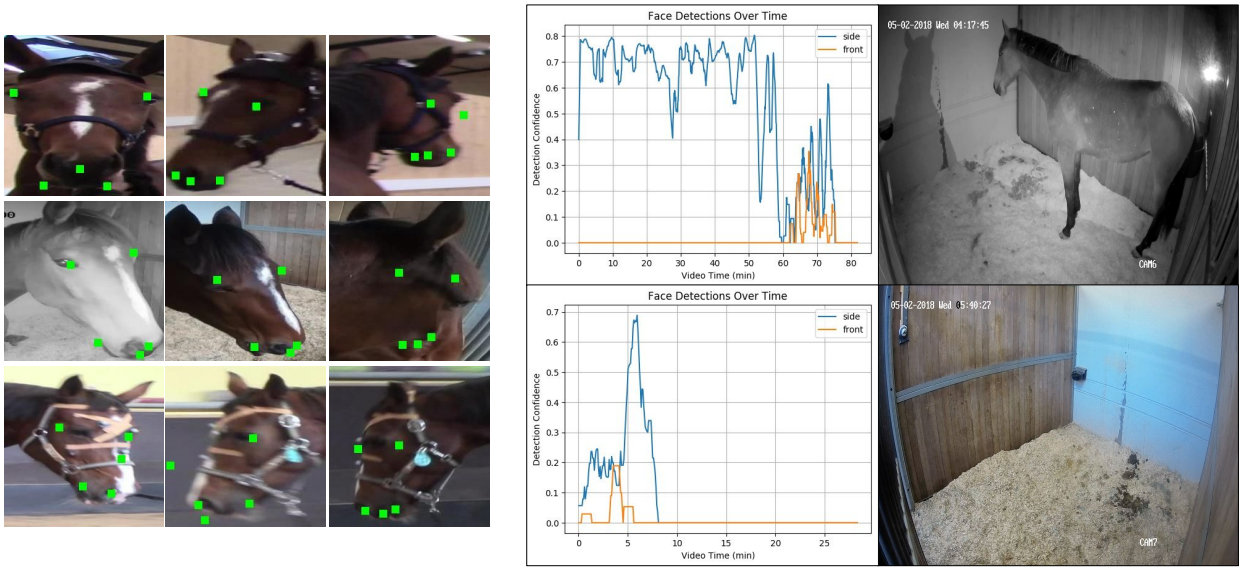

Figure 3.3: **Left:** Examples of keypoint detection on frames from three videos. Incorrect detections correspond to frames where the horse face is not clearly visible or occluded. Note that all keypoints even when those parts are not visible. **Right:** By automatically identifying videos with annotatable segments, our tool prioritizes a video where the horse face is visible and still for majority of the video (top), over a video where the horse is not present in the stall past the first few minutes (bottom).

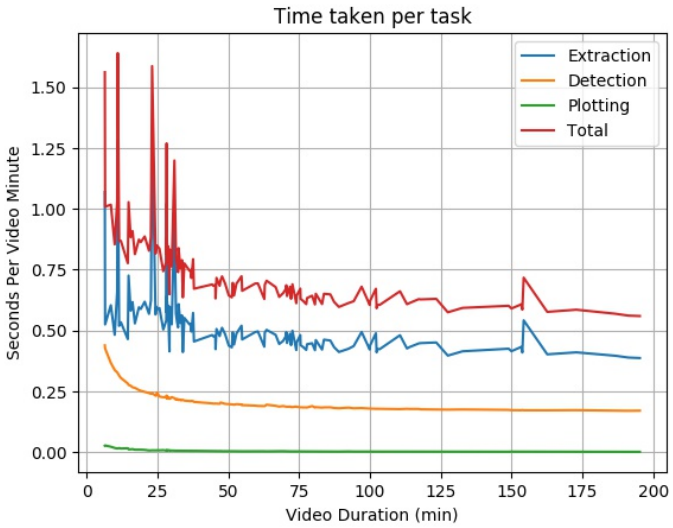

Figure 3.4: Seconds per video minute taken to perform each task in our approach for videos of different total length.

	Frame Extraction	Face Detection	Saving & Plotting	Total Time
Short Video 0:06:28	1.06	0.44	0.03	1.56
Long Video 3:15:15	0.39	0.17	0.001	0.56
Average	0.54± 0.13	0.21± 0.04	0.006± 0.004	0.77± 0.16

Figure 3.5: The seconds per video minute taken for the shortest and longest videos (duration in H:MM:SS format). The average times taken are shown in the last row. Our method is more efficient for longer videos.

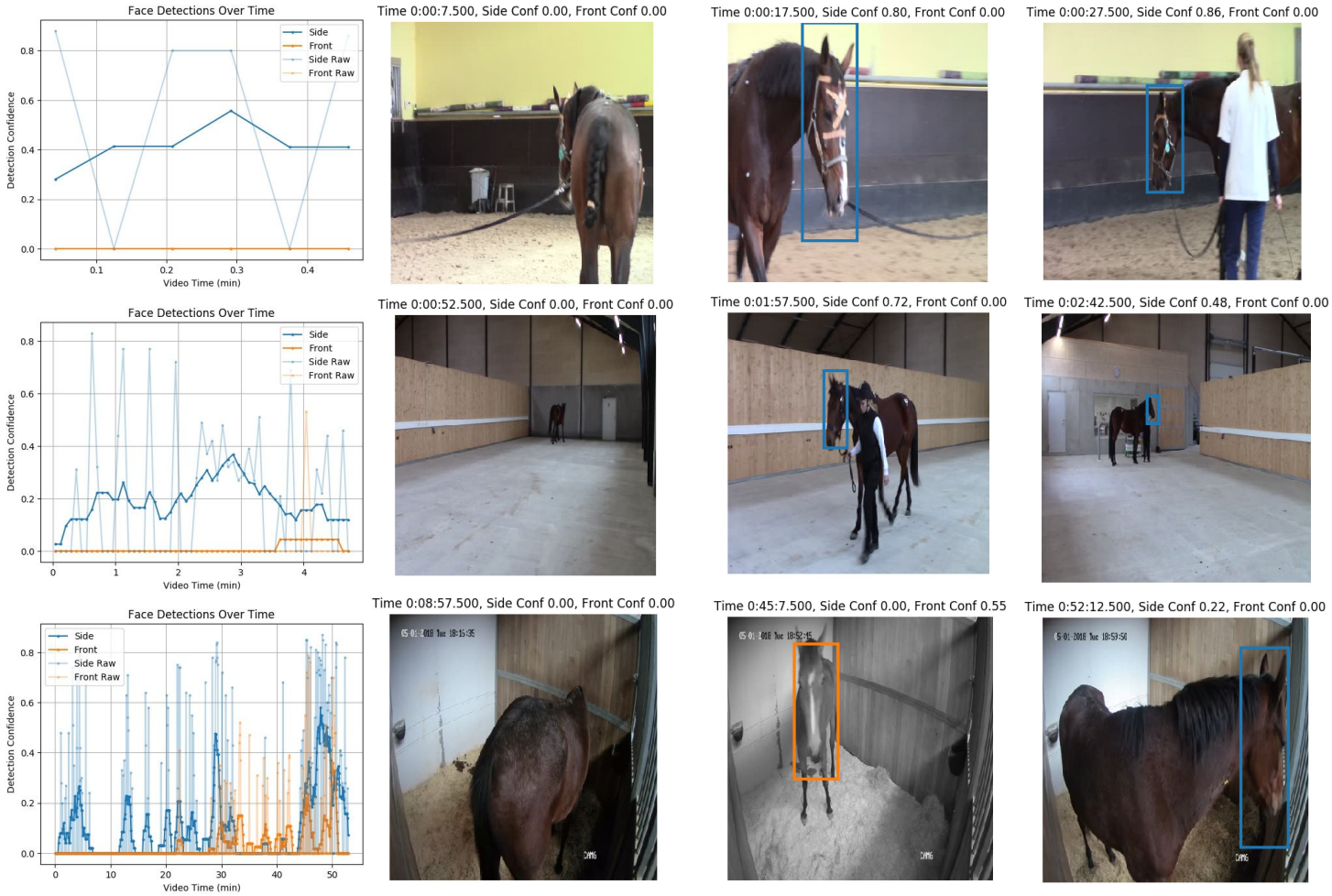

Figure 3.6: **Face finder detections**. The first and second videos show horses trotting in a circle. Detection peaks correspond to times when the horse face is visible, with no detections when the horse face is self-occluded. Consistently high detections around the three minute mark for the second video correspond to horse standing still in side view. Detections are correct even in night-view camera mode, or when the horse is in an unusual position relative to camera (bottom row).

Chapter 4

Interspecies Knowledge Transfer for Facial Keypoint Detection

The last chapter presented a means of horse face detection and its practical application in assisting EquiFACS coders. Beyond faces, successful automatic analysis of facial expressions, as well as FACS detection, often relies on the accurate and automatic localization of facial parts such as the eye centers. The detection of such meaningful facial parts, or keypoints, is the subject of this chapter.

Facial keypoint detection is a necessary precondition for face alignment and registration, and impacts facial expression analysis, facial tracking, as well as graphics methods that manipulate or transform faces. While human facial keypoint detection is a mature area of research, despite its importance, animal facial keypoint detection is a relatively unexplored area. For example, veterinary research has shown that horses [17, 16], mice [64], sheep [88], and cats [89] display facial expressions of pain – a facial keypoint detector could be used to help automate such animal pain detection. In this work, we tackle the problem of facial keypoint detection for animals, with a focus on horses and sheep.

Convolutional neural networks (CNNs) have demonstrated impressive performance for *human* facial keypoint detection [90, 91, 92, 93, 94, 95, 96, 97], which makes CNNs an

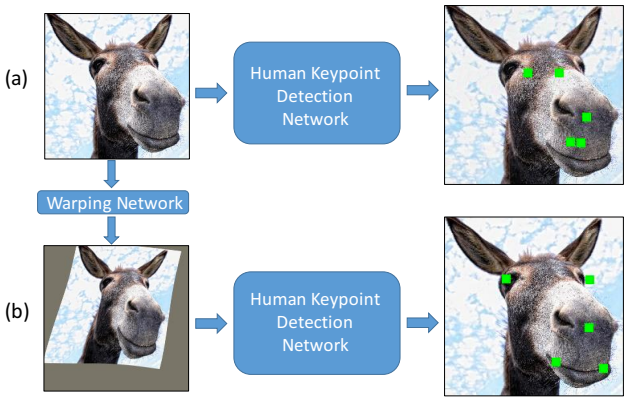

Figure 4.1: **Main idea.** (**a**) Directly finetuning a human keypoint detector to horses can be suboptimal, since horses and humans have very different shapes and appearances. (**b**) By warping a horse to have a more human-like shape, the pre-trained human keypoint detector can more easily adapt to the horse's appearance.

attractive choice for learning facial keypoints on animals. Unfortunately, training a CNN from scratch typically requires large amounts of labeled data, which can be time-consuming and expensive to collect. Furthermore, while a CNN can be finetuned when there is not enough training data for the target task, a pre-trained network's extent of learning is limited both by the amount of data available for fine-tuning, as well as the *relatedness of the two tasks*. For example, previous work demonstrate that a network trained on man-made objects has limited ability to adapt to natural objects [98], and additional pretraining data is only beneficial when related to the target task [99].

While there are large datasets with human facial keypoint annotations (e.g., AFLW has ∼26000 images [71]), there are, unfortunately, no large datasets of animal facial keypoints that could be used to train a CNN from scratch (e.g., the sheep dataset from [78] has only ∼600 images). At the same time, the structural differences between a human face and an animal face means that directly fine-tuning a human keypoint detector to animals can lead to a sub-optimal solution (as we demonstrate in Sec. 4.3).

In this chapter, we address the problem of transferring knowledge between two different types of data (human and animal faces) for the same task (keypoint detection). How can we achieve this with a CNN? Our key insight is that rather than adapt a pre-trained network

to training data in a new domain, we can first do the *opposite*. That is, *we can adapt the training data from the new domain to the pre-trained network*, so that it is better conditioned for finetuning. By mapping the new data to a distribution that better aligns with the data from the pre-trained task, we can take a pre-trained network from the loosely-related task of human facial keypoint detection and finetune it for animal facial keypoint detection. Specifically, our idea is to explicitly warp each animal image to look more human-like, and then use the resulting warped images to finetune a network pre-trained to detect human facial keypoints. See Fig. 4.1.

Intuitively, by warping animal faces to look more human-like we can correct for their shape differences, so that during finetuning the network need only adapt to their differences in appearance. For example, the distance between the corners of a horse's mouth is typically much smaller than the distance between its eyes, whereas for a human these distances are roughly similar – a shape difference. In addition, horses have fur, and humans do not – an appearance difference. Our warping network adjusts for the shape difference by stretching out the horse's mouth corners, while during finetuning the keypoint detection network learns to adjust for the appearance difference.

Contributions. Our contributions are three fold: First, we introduce a novel approach for animal facial keypoint detection that transfers knowledge from the loosely-related domain of human facial keypoint detection. Second, we provide a new annotated horse facial keypoint dataset consisting of 3717 images. Third, we demonstrate state-of-the-art results on keypoint detection for horses and sheep. By transforming the animal data to look more human-like, we attain significant gains in keypoint detection accuracy over simple finetuning. Importantly, the gap between our approach and simple finetuning widens as the amount of training data is reduced, which shows the practical applicability of our approach to small datasets. Our data and code are available at https://github.com/menoRashid/animal_human_kp.

4.1 Related work

Facial landmark detection and alignment are mature topics of research in computer vision. Classic approaches include Active Appearance Models [100, 101, 102, 103], Constrained Local Models [104, 105, 106, 107], regression based methods [108, 109, 110, 111] with a cascade [112, 113, 114], and an ensemble of exemplar based models [115]. Recent work extends cascaded regression models by learning predictions from multiple domain-specific regressors [116] or by using a mixture of regression experts at each cascade level [117]. These models also demonstrate good performance when solved simultaneously with a closely related task, such as face detection [118], 3D face reconstruction [119], and facial action unit activation detection [120].

In the deep learning domain, coarse-to-fine approaches refine a coarse estimate of keypoints through a cascade [121, 122, 123, 97] or with branched networks [124]. Others assist keypoint detection by using separate cluster specific networks [125], augmenting it with related auxiliary tasks [126], initializing with head pose predictions [127], correcting for deformations with a spatial transformer [96], incorporating shape basis and thin plate spline transformations [128], formulating keypoint detection as a dense 3D face model fitting problem [94, 95], or using deep regression models in combination with de-corrupt autoencoders [93]. Recent work explore using recurrent neural networks [90, 91, 92].

While deep learning approaches demonstrate impressive performance, they typically require large annotated datasets. Rather than collect a large dataset, [129] uses domain specific augmentation techniques to *synthesize* pose, shape, and expression variations. However, it relies on the availability of 3D face models, and addresses the related but separate problem of face recognition. Similarly, [31] leverages large datasets available for face recognition to train a deep network, which is then used to guide training of an expression recognition network using only a small amount of data. However, while [31] transfers knowledge between two different tasks (face recognition and expression recognition) that rely on the same type of data (human faces), we transfer knowledge between two different data sources (human and

animal faces) in order to solve the same task (facial keypoint detection).

To the best of our knowledge, *facial* keypoint detection in animals is a relatively unexplored problem. Very recently, [78] proposed an algorithm for keypoint detection in sheep, using triplet interpolated features in a cascaded shape regression framework. Unlike our approach, it relies on hand-crafted features and does not transfer knowledge from human to animal faces. Keypoint localization on birds has been explored in [130, 131, 132, 133], though these approaches do not focus on facial keypoint detection.

4.2 Approach

Our goal is to detect facial keypoints in animals without the aid of a large annotated animal dataset. To this end, we propose to adapt a pre-trained *human* facial keypoint detector to *animals* while accounting for their interspecies domain differences. For training, we assume access to keypoint annotated animal faces, and keypoint annotated human faces and their corresponding pre-trained human keypoint detector. For testing, we assume access to an animal face detector (i.e., we focus only on facial keypoint detection and not face detection).

Our approach has three main steps: (1) finding nearest neighbor human faces that have similar pose to each animal face; (2) using the nearest neighbors to train an animal-to-human warping network; and (3) using the warped (human-like) animal images to fine-tune a pre-trained human keypoint detector for animal facial keypoint detection.

4.2.1 Nearest neighbors with pose matching

In order to fine-tune a (loosely-related) human facial keypoint detector to animals, our idea is to first warp the animal faces to have a more human-like shape so that it will be easier for the pre-trained human detector to adapt to the animal data. One challenge is that an arbitrary animal and human face pair can exhibit drastically different poses (e.g., a right-facing horse and a left-facing person), which can making warping extremely challenging or

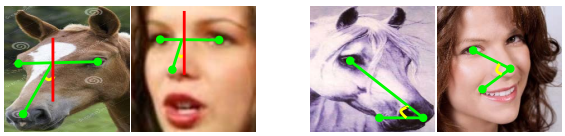

Figure 4.2: We approximate facial pose using the angle generated from the keypoint annotations. The keypoints used to compute the angle-of-interest depend on which facial parts are visible. For example, on the right, the horse's right eye and right mouth corner are not visible, so the three keypoints used are the left eye, nose, and left mouth corner. While simple, we find this approach to produce reliable pose estimates.

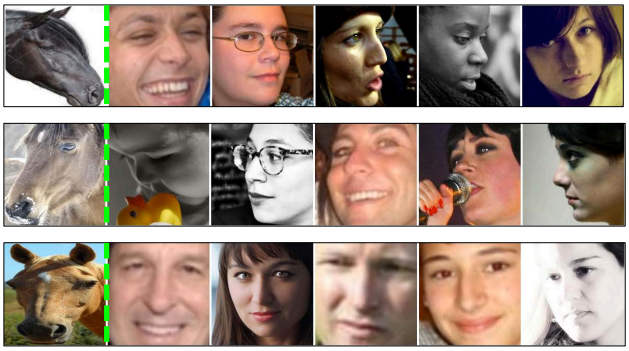

Figure 4.3: For each animal image (1st column), we find the nearest human neighbors in terms of pose. These human neighbors are used to train a warp network that warps an animal to have human-like face shape.

even impossible. To alleviate this difficulty, we first find animals and humans that are in similar poses.

If we had pose classifiers/annotations for both animal and human faces, then we could simply use their classifications/annotations to find compatible animal and human pairs. However, in this work, we assume we do not have access to pose classifiers nor pose annotations. Instead, we *approximate* a face pose given its keypoint annotations. More specifically, we compute the angular difference between a pair of human and animal keypoints, and then pick the nearest human faces for each animal instance.

For each animal training instance A_i, we find its nearest human neighbor training instance H_{j*} based on pose:

$$nn(A_i) = H_{j*} = \underset{H_j}{\operatorname{argmin}} \, |\measuredangle^* A_i - \measuredangle^* H_j|, \tag{4.1}$$

where j indexes the entire human face training dataset, and the angle of interest $\measuredangle^*$ is measured in two different ways depending on the animal face's visible keypoints. When both eyes and the nose are present, we use $\measuredangle^* = \measuredangle N E_c V$, where E_c is the midpoint between the eye centers, N is the nose position, and V is a vertical line centered at E_c. If only the left eye is visible, then we use the left eye, nose, and left mouth keypoints: $\measuredangle^* = \measuredangle E_l N M_l$ (and $\measuredangle E_r N M_r$ if the right eye is visible). These cases are illustrated in Fig. 4.2.

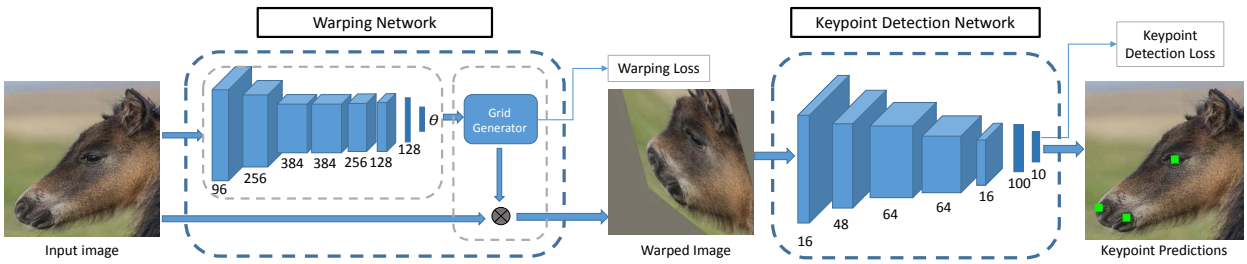

Figure 4.4: Our network architecture for animal facial keypoint detection. During training, the input image is fed into the warping network, which is directly supervised using keypoint-annotated human and animal image pairs with similar pose. The warping network warps the input animal image to have a human-like shape. The warped animal face is then passed onto the keypoint detection network, which finetunes a pre-trained human keypoint detection network with the warped animal images. During testing, the network takes the input image and produces 5 keypoint predictions for left eye, right eye, nose, left mouth corner, and right mouth corner.

While simple, we find this approach to produce reliable pose estimates. In our experiments, we find the $K = 5$ nearest human neighbors for each animal face. Fig. 4.3 shows some examples. Since we use the TPS transformation for warping animals to humans (as described in the next section), we only compute matches for animal faces with at least three keypoints and ignore human matches whose keypoints are close to colinear, which can cause gross artifacts in warping. Note that we do not do pose matching during testing, since we do not have access to ground-truth keypoints; instead we rely on the ensuing warping network to have learned the "right" warp for each animal face pose (based on its appearance) during training.

4.2.2 Interspecies face warping network

Now that we have the nearest human faces (in terms of pose) for each animal face, we can use these matches to train an animal-to-human face warping network. This warping network serves to adapt the shape of the animal faces to more closely resemble that of humans, so that a pre-trained human facial keypoint detector can be more easily fine-tuned on animal faces.

For this, we train a CNN that takes as input an animal image and warps it via a thin plate

spline (TPS) [134] transformation. Our warping network is a spatial transformer [135], with the key difference being that our warps are directly supervised, similar to [96].[1] Our network architecture is similar to the localization network in [136]; it is identical to Alexnet [20] up to the fifth convolutional layer, followed by a 1×1 convolution layer that halves the number of filters, two fully-connected layers, and batch normalization before every layer after the fifth. During training, the first five layers are pre-trained on ImageNet. We find these layer/filter choices to enable good TPS transformation learning without overfitting. See Fig. 4.4 (left).

For each animal and human training image pair, we first calculate the ground-truth TPS transformation using its corresponding keypoint pairs and apply the transformation to produce a ground-truth warped animal image. We then use our warping network to compute a predicted warped animal image. To train the network, we regress on the difference between the ground-truth warped image and predicted warped image pixel position offsets, similar to [137]. Specifically, we use the squared loss to train the network:

$$L_{warp}(A_i) = \sum_m (p_{i,m}^{pred} - p_{i,m}^{gt})^2, \tag{4.2}$$

where A_i is the i-th animal image, $p_{i,m}^{pred}$ and $p_{i,m}^{gt}$ are the predicted offset and ground-truth offset, respectively, for pixel m.

It is important to note that our warping network requires no additional annotation for training, since we only use the animal/human keypoint annotations to find matches (which are already available and necessary for training their respective keypoint detectors). In addition, since each animal instance has multiple ($K = 5$) human matches, the warping network is trained to identify multiple transformations as potentially correct. This serves as a form of data augmentation, and helps make the network less sensitive to outlier matches.

[1] In contrast, in [135] the supervision only comes from the final recognition objective e.g., keypoint detection. We show in Sec. 4.3 that direct warping supervision produces superior performance.

4.2.3 Animal keypoint detection network

Our warping network from the previous section conditions the distribution of the animal data to more closely resemble human data, so that we can harness the large *human* keypoint annotated datasets that are readily available for *animal* keypoint detection. The final step is to finetune a pre-trained human facial keypoint detection network to detect facial keypoints on our warped animal faces.

Our keypoint detector is a variant of the Vanilla CNN architecture used in [125]. The network has four convolutional layers, and two fully-connected layers with absolute tanh non-linearity, and max-pooling in the last three convolutional layers. We adapt it to work for larger images—we use 224×224 images as input rather than 40×40 used in [125]—by adding an extra convolutional and max-pooling layer. In addition, we add batch normalization after every layer since we find the tanh layers in the original network to be prone to saturation. Fig. 4.4 (right) shows the architecture. Our keypoint detection network is pre-trained on human facial keypoints on the AFLW [71] dataset and the training data used in [121] (a total of 31524 images).

To finetune our keypoint network, we use the smooth $L1$ loss (equivalent to the Huber loss with δ=1) used in [138] since it is less sensitive to outliers that may occur with unusual animal poses:

$$L_{keypoint}(A_i) = \sum_n smooth_{L_1}(k_{i,n}^{pred} - k_{i,n}^{gt}), \tag{4.3}$$

where A_i is the i-th animal image, $k_{i,n}^{pred}$ and $k_{i,n}^{gt}$ are the predicted and ground-truth keypoint position, respectively, for the n-th keypoint, and $smooth_{L_1}$ is

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \tag{4.4}$$

We set the loss for predicted keypoints with no corresponding ground-truth annotation (due to occlusion) to zero.

4.2.4 Final architecture

In our final model, we fit the warping network before a keypoint detection network that is pre-trained on human keypoint detection. We use the two losses to jointly finetune both networks. The keypoint detection loss $L_{keypoint}$ (Eqn. 4.3) is back propagated through both the keypoint detection network, as well as the warping network. Additionally, the warping loss L_{warp} (Eqn. 4.2) is backpropagated through the warping network, and the gradients are accumulated before the weights for both networks are updated. See Fig. 4.4.

In the testing phase, our keypoint network predicts all 5 facial keypoints for every image. In our experiments, we do not penalize the network for keypoint predictions that are not visible in the image and results are reported only for predicted keypoints that have corresponding ground-truth annotation. For evaluation, the keypoints predicted on warped images are transferred back to the original image using the TPS warp parameters.

4.2.5 Horse Facial Keypoint dataset

As part of this work, we created a new horse dataset to train and evaluate facial keypoint detection algorithms. We collected images through Google and Flickr by querying for "horse face", "horse head", and "horse". In addition, we included images from the PASCAL VOC 2012 [77] and Imagenet 2012 [139] datasets. There are a total of 3717 images in the dataset: 3531 for training, and 186 for testing. We annotated each image with face bounding boxes, and 5 keypoints: left eye center, right eye center, nose, left mouth corner, and right mouth corner.

4.3 Experiments

In this section, we analyze our model's keypoint detection accuracy, and perform ablation studies to measure the contribution of each component. In addition, we evaluate our method's performance as the amount of training data is varied, and also measure an upper-

bound performance if animal-to-human warping were perfect.

Baselines. We compare against the algorithm presented in [78], which uses triplet-interpolated features (TIF) in a cascaded shape regression framework for keypoint detection on animals. We also develop our own baselines. The first baseline is our full model without the warping network. It simply finetunes the pre-trained human facial keypoint network on the animal dataset ("BL FT"). The second baseline is our full model without the warping loss; i.e., it finetunes the pre-trained human facial keypoint network and the warping network with only the keypoint detection loss. This baseline is equivalent to the spatial transformer setting presented in [135]. We show results for this with TPS warps ("BL TPS"). The third baseline trains the keypoint detection network from scratch; i.e., without any human facial keypoint detection pretraining and without the warping network ("Scratch").

Datasets. We pretrain our keypoint detection network on human facial keypoints from the AFLW [71] dataset and the training data used in [121] (a total of 31524 images). This dataset is also used for animal to human nearest neighbor retrieval. We evaluate keypoint detection on two animals: horses and sheep. For the horse experiments, we use our Horse Facial Keypoint dataset, which consists of 3531 images for training and 186 for testing. For the sheep experiments, we manually annotated a subset of the dataset provided in [78] with mouth corners so that we have the same 5 keypoints present in the human dataset. The dataset consists of 432 images for training and 99 for testing.

Evaluation metric. We use the same metric for evaluation as [78]: If the euclidean distance between the predicted and ground-truth keypoint is more than 10% of the face (bounding box) size, it is considered a failure. We then compute the average failure rate as the percentage of testing keypoints that are failures.

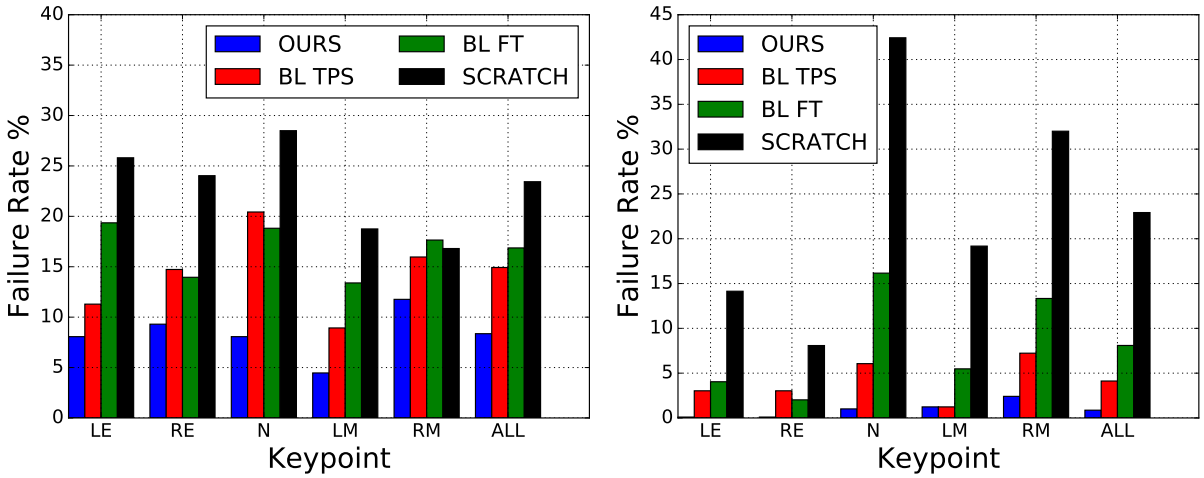

Figure 4.5: Average keypoint detection failure rate (% of predicted keypoints whose euclidean distance to the corresponding ground-truth keypoint is more than 10% of the face bounding box size). Horses (**left**) and Sheep (**right**). Our approach outperforms the baselines. *Lower is better.* See text for details.

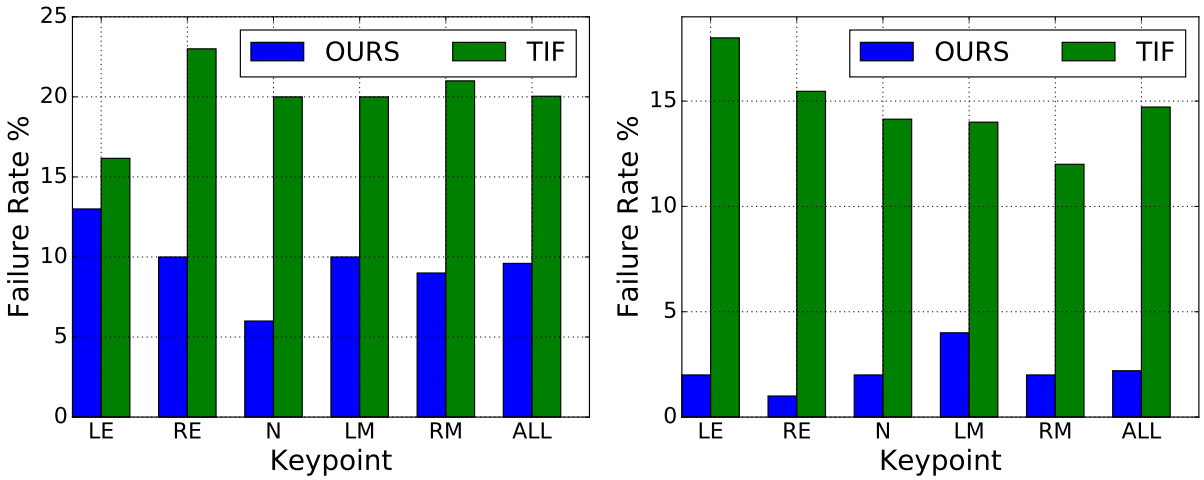

Figure 4.6: Average keypoint detection failure rate for Horses (**left**) and Sheep (**right**). Our approach significantly outperforms the Triplet Interpolated Features (TIF) approach of Yang et al. [78], which combines hand-crafted features with cascaded shape regressors. *Lower is better.*

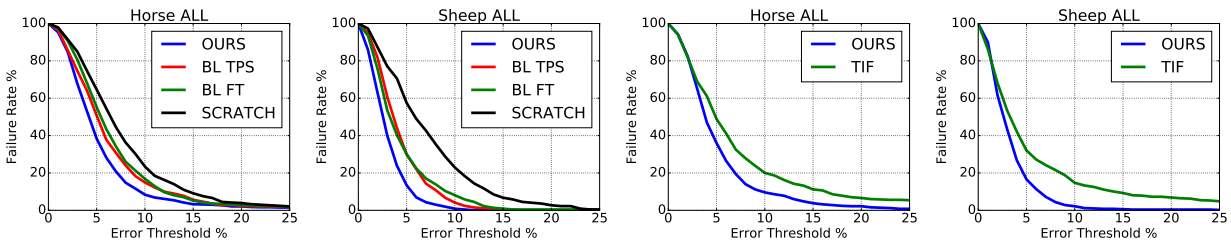

Figure 4.7: Average keypoint detection failure rate across all keypoints for our system vs. our baselines (first two plots) and the Triplet Interpolated Features (TIF) approach of Yang et al. [78] (last two plots). Our system sustains lower failure rates across stricter failure thresholds than all baselines.

Training and implementation details. We find that pretraining the warping network before joint training leads to better performance. To train the warping and keypoint network, we use $K = 5$ human neighbors for each animal instance. These matches are also used to supervise the "GT Warp" network described in Sec. 4.3.4.

For the TPS warping network, we use a 5×5 grid of control points. We optimize all networks using Adam [140]. The base learning rate for the warp network training is 0.001, with a $\frac{1}{10}\times$ lower learning rate for the pre-trained layers. It is trained for 50 epochs, with the learning rate lowered by $\frac{1}{10}\times$ after 25 epochs. During full system training, the warp network has the same learning rates, while the keypoint detection network has a learning rate of 0.01. We train the network for 150 epochs, lowering the learning rate twice after 50 and 100 epochs. Finally, we use horizontal flips and rotations from $-10°$ to $10°$ at increments of $5°$ for data augmentation.

4.3.1 Comparison with our baselines

We first compare our full model with our model variant baselines. Figure 4.5 (left) and (right) show results on horse and sheep data, respectively. We outperform all of our baselines significantly for both horses and sheep, with an average failure rate across keypoints at 8.36% and 0.87%, respectively.

Overall, the failure rate for all methods (except Scratch) for sheep is lower than that for

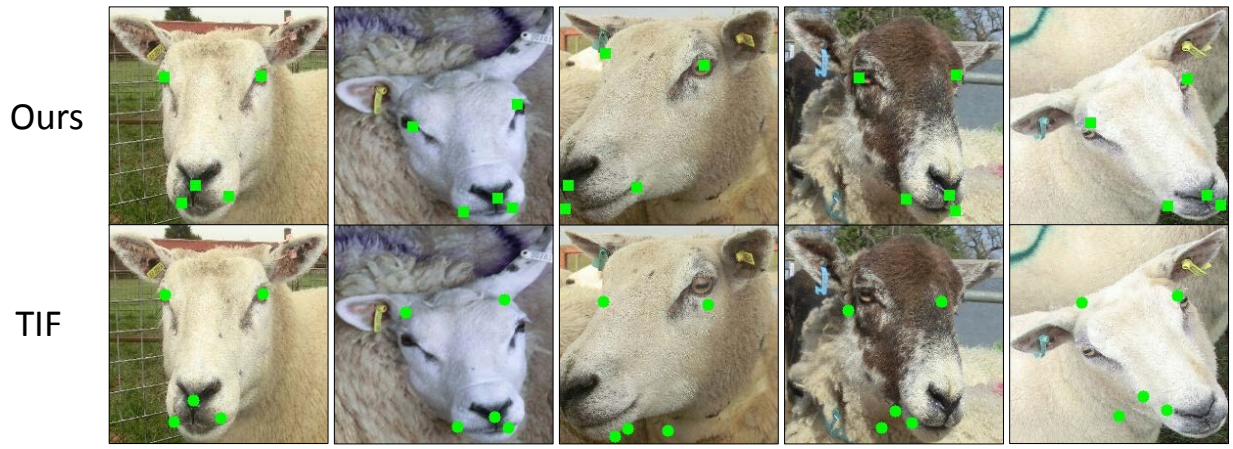

Figure 4.8: Qualitative examples comparing our approach and Yang et al. [78] on their Sheep dataset. While [78] can produce good predictions (first column), overall, our method produces significantly more accurate results.

horses. The main reason is due to the pose distribution of human and sheep data being more similar than that of human and horse data. The human and sheep data have 72% and 84% of images in frontal pose (faces with all 5 keypoints visible) as compared to only 29% for horses. The majority (60%) of horse faces are side-view (faces with only 3 keypoints visible). This similarity makes it easier for the human pre-trained network to adapt to sheep than to horses. Nonetheless, the fact that our method outperforms the baselines for both datasets, demonstrates that our idea is generalizable across different types of data.

These results also show the importance of each component of our system. Training with a human pre-trained network does better than training from scratch (BL FT vs. Scratch); adding a warping network that is only *weakly-guided* by the keypoint detection loss further improves results (BL TPS vs. BL FT); and finally, directly supervising the warping network to produce animal faces that look more human-like leads to the best performance (Ours vs. BL TPS). The first two plots in Fig. 4.7 show the results of varying the acceptance threshold (on the euclidean distance between the ground-truth and predicted keypoint) for a valid keypoint on our and the baselines' performance. Our method sustains superior accuracy across thresholds, which again indicates that we predict keypoints more accurately.

Fig. 4.11 shows qualitative examples of predicted keypoints and predicted warps for ours

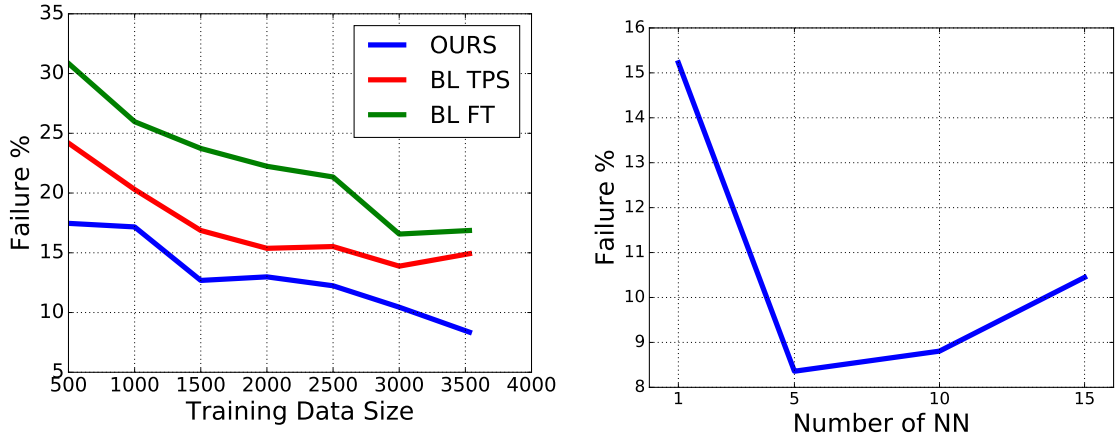

Figure 4.9: (**left**) Average keypoint detection failure rate as a function of the number of training instances on the Horse dataset. Our failure rate increases more gracefully compared to the baselines as the number of training images is decreased. *Lower is better.* (**right**) Increasing the number of human face neighbors for an animal face instance increases performance until noisy neighbors cause performance to drop.

	GT Warp	Ours
Failure Rate %	7.76%	8.36%

Figure 4.10: Average keypoint detection failure rate across all keypoints on the Horse dataset, comparing our approach to an upper-bound ground-truth warping baseline. *Lower is better.*

and the baselines. Noticeably, the TPS warps produced without the warping loss (BL TPS Warp) fail to distinguish between the different horse poses, and also do not warp the horse faces to look more human like. On the other hand, our warping network is able to do both tasks well since it is directly supervised by pose specific human matches. By warping the horses to have more human-like shape, our method produces more precise keypoint predictions than the baselines. The last two rows show typical failure examples due to extreme pose or occlusion.

4.3.2 Comparison with Yang et al.

We next compare our method to the Triplet Interpolated Features (TIF) approach of [78], which is the state-of-the-art animal keypoint detector. The method requires the existence of all landmarks in all training examples. We therefore picked a subset of the horse and sheep

images where all 5 keypoints are visible and marked: 345/100 train/test images for sheep, and 982/100 train/test images for horses.

Fig. 4.8 shows qualitative examples comparing our method's keypoint predictions vs. those made by TIF. TIF often fails to handle large appearance and pose variations. This is also reflected in the quantitative results, which are shown in Fig. 4.7 (third) and Fig. 4.6 (left) for the horse dataset and Fig. 4.7 (fourth) and Fig. 4.6 (right) for the sheep dataset. We significantly outperform TIF on both datasets (10.44% and 12.52% points lower failure rate for horses and sheep, respectively). The main reason is because we use a high capacity deep network, whereas TIF is a shallow method that learns with hand-crafted features. Importantly, the reason that we are able to use such a high capacity deep network—despite the limited training data of the animal datasets—is precisely because we correct for the shape differences between animals and humans in order to *finetune* a pre-trained human keypoint detection network.

4.3.3 Effect of training data size

In this section, we evaluate how the performance of our network changes as the amount of training data varies. For this, we train and test multiple versions of our model and the baselines, each time using 500 to 3531 training images in 500 image increments on the Horse dataset.

Figure 4.9 (left) shows the result. While the performance of all methods decreases with the training data amount, our performance suffers much less than that of the simple finetuning and TPS baselines. In particular, when using only 500 training images, our method has a 6.72% point lower failure rate than the TPS baseline while relying on the same network architecture, and a 13.39% point lower failure rate than simple finetuning, *without using any additional training data or annotations.*

This result demonstrates that our algorithm adapts well to small amounts of training data, and bolsters our original argument that explicitly correcting for interspecies shape

differences enables better finetuning, since the pre-trained human keypoint detection network can mostly focus on the appearance differences between the two domains (humans and animals). Importantly, it also shows the practical applicability of our approach to small datasets.

4.3.4 Effect of warping accuracy

We next analyze the influence of warping accuracy on keypoint detection. For this, we first analyze the performance of our keypoint detection network when finetuned with *ground-truth* warped images ("GT Warp"), where we use the ground-truth keypoint annotations between human and horse faces for warping (i.e., the keypoint detection network is finetuned with ground-truth warped images). In a sense, this represents the upper bound of the performance of our system.

Table 4.10 shows the results on our Horse dataset. First, the GT Warp upper-bound produces even lower error rates than our method, which demonstrates the efficacy of the idea of correcting for shape differences by warping. At the same time, the non-negligible error rate of GT Warp also hints at the limitation of our warping network's training data and/or pose matching strategy. Better training data, with either a different algorithm for nearest pose neighbor matching or an increase in the keypoints that are annotated could potentially lead to a better upper-bound, and would likely provide improvements for our approach as well.

4.3.5 Evaluation of Nearest Neighbors

Finally, we evaluate the importance of human nearest neighbors for our system. We vary the number of nearest neighbors used for training our full system from $K = 1$ to $K = 15$ at increments of 5 for our full Horse training set. The result is shown in Figure 4.9 (right). While the error rate decreases as the number of neighbors used for training is increased in the beginning, eventually, the noise in retrieved nearest neighbors causes the error rate to

increase.

4.4 Discussion

We presented a novel approach for localizing facial keypoints on animals. Modern deep learning methods typically require large annotated datasets, but collecting such datasets is a time consuming and expensive process.

Rather than collect a large annotated animal dataset, we instead warp an animal's face shape to look like that of a human. In this way, our approach can harness the readily-available human facial keypoint annotated datasets for the loosely-related task of animal facial keypoint detection. We compared our approach with several strong baselines, and demonstrated state-of-the-art results on horse and sheep facial keypoint detection. Finally, we introduced a novel Horse Facial Keypoint dataset, which we hope the community will use for further research on this relatively unexplored topic of animal facial keypoint detection.

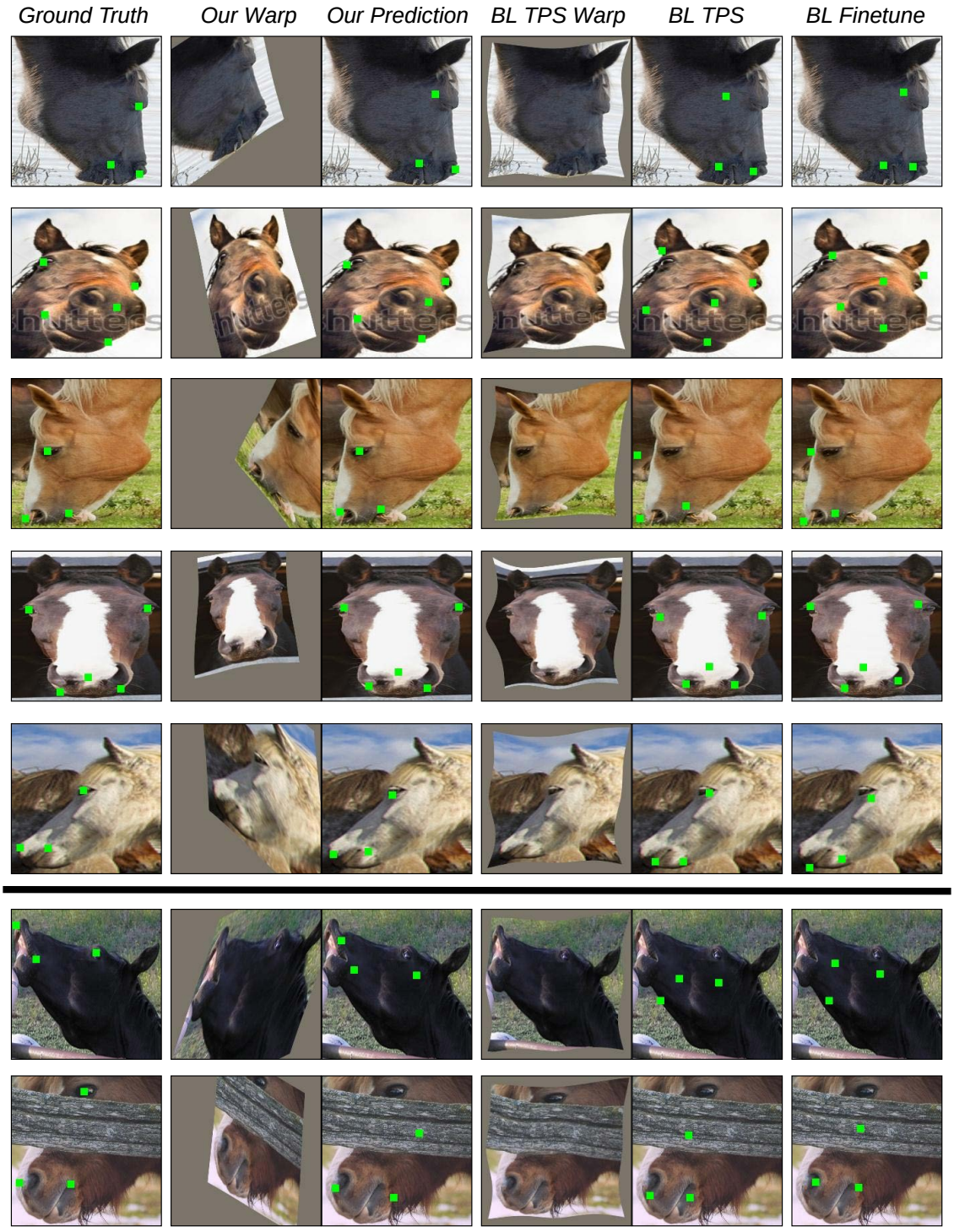

Figure 4.11: Qualitative examples of predicted keypoints and predicted warps for ours and the baselines. The first five rows show examples where our method outperforms the baseline. While the baselines also produce reasonable results, by warping the horses to have more human-like shape, our method produces more precise keypoint predictions. For example, in the first row, the baselines do not localize the nose and mouth corner as well as ours. The last two rows show typical failure examples due to extreme pose or occlusion.

Chapter 5

Action Graphs: Weakly-supervised Action Localization with Graph Convolution Networks

In the previous chapter we addressed the problem of facial keypoint detection which is an important prerequisite for facial expression classification in a fully supervised setting. At the same time, it is important to recognize that creating an expression dataset that enables fully supervised machine learning is a difficult and time consuming task. Given videos of horses, expert annotators would have to mark the time points when the horse starts and stops making a pain face, and would at minimum take the duration of the video length for each feature that needs to be marked. Alternatively, expert annotators could provide weak labels: a video can be labeled as a pain video if at any point in it the horse expresses pain, and a no pain video otherwise. These less informative, but easier to obtain labels can then be used to train a pain detection model.

In computer vision this problem maps exactly to the problem of weakly supervised temporal activity localization.Temporal activity localization is the problem of identifying the start and end times of every action's occurrence [141, 142]. In a fully supervised setting,

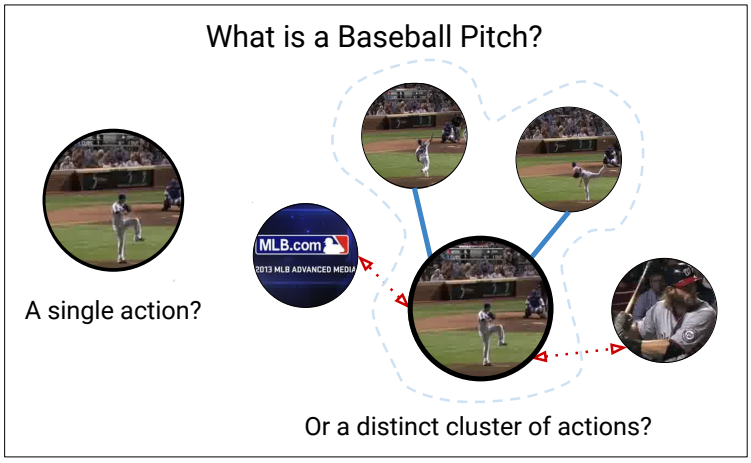

Figure 5.1: **Key Idea:** A baseball pitch is not defined by a single action – rather it is defined by a series of smaller actions that are distinct from other actions in a video. Despite this, prior methods classify every time segment individually before collating predictions for localization (left). We instead explicitly model what each segment is similar to – blue edges – and different from – red edges – for weakly-supervised temporal action localization (right).

every training video is annotated with the start and end time of each action's occurrence. However, acquiring manual temporal annotations is an onerous task and severely limits both the number and diversity of actions that a system can be trained to identify. In contrast, systems that can successfully classify and temporally localize actions with *weak* video-level labels—that only state whether an activity is present in the video or not—provide a more scalable solution.

Without frame-level annotations, weakly-supervised systems must rely on similarity cues between video time segments. Specifically, they must (1) use the dissimilarity between foreground segments of different action classes to classify videos correctly; (2) use the similarity/relationship between foreground segments of the same action to determine its full extent; and (3) infer that the similarity between segments of different actions' videos are indicative of background segments.

Although great progress has been made on this challenging problem, existing approaches [143, 144, 145, 146] do not explicitly model the *relationships between time segments* to inform their final predictions. Instead, most approaches first split the video into multiple time segments, and classify each segment separately. These segment-level predictions are then

pooled to perform the final video-level classification using multiple instance learning [147]. The relationships between time segments are either only implicitly used during training to learn attention [145], perform the final video-level classification [146], or to create good features [143, 144], but are not used during test time. In contrast, Xu et al. [148] use a recurrent neural network to model relationships between time segments. However, similarity between time segments that are temporally distant, or belong to different videos cannot be modeled in their framework. In other words, the model lacks the ability to ensure that all time segments regardless of temporal location that are related to the same action are treated similarly.

Main idea. Our main idea is to explicitly model the similarity relationships between time segments of videos in order to classify and localize actions in videos. We use graph convolution networks (GCNs) [149] for this purpose.

Similar to regular convolution networks, GCNs also perform nonlinear transformations on the input features. However, in addition, GCNs treat input features as nodes in a graph with weighted edges. By setting the edge weights to be proportional to the level of similarity between nodes, GCNs allow feature similarity and dissimilarity to be incorporated into the weight learning process as gradients are propagated across weighted edges, as well as during test time as inference is performed over an entire graph.

By using GCNs, our method explicitly ensures that relationships between time segments are considered during both training and testing. We represent each segment in a video as a node in a graph, and edges between nodes are weighted by their similarity. Each segment's feature representation is transformed to a weighted average of all segments it is connected to, with weights based on learned edge strength. These weighted average features are then used to learn a multiple instance learning based video classifier. We use appearance and motion similarity between segments to determine edge weights: two nodes that have similar RGB and optical flow features have a stronger edge between them than two nodes that have dissimilar RGB and optical flow features. In this way, the learned weights operate on

groups of features together, rather than on individual time segments. This helps prevent the network from focusing on just a few discriminative parts of the video.

Contributions. (1) A novel graph convolution approach for weakly-supervised action localization. Our method is based on an appearance and motion similarity graph and is the first to use graph convolutions in the weakly-supervised action localization setting. (2) We analyze each component of our model, explore other graph based alternatives, and quantitatively and qualitatively compare against other non-graph based approaches. (3) We push the state-of-the-art on widely-used action detection datasets in the weakly-supervised setting - THUMOS'14 [150] and ActivityNet 1.2 [151], and are the first to present results on Charades [152].

5.1 Related work

Weakly supervised action localization has many different variants in literature. [144] encourages time segments with similar classification predictions to have similar intermediate deep features using a Co-Activity Similarity Loss. Like us, it uses feature similarity between segments to improve localization. However, unlike our approach, it exclusively uses feature similarity to provide training supervision, and does not model feature relationships to make predictions. Others discourage the network from focusing only on the most discriminative time segments via random hiding [153], or their iterative removal during training [154]. While [155] uses a contrastive loss for temporally fuller localizations, [156] additionally uses a coherence loss for visually consistent action identification. More recent works learn to attend and pool per time segment predictions during training [145, 143], while UntrimmedNets [146] simultaneously learns to classify and select the most salient segments in a video. However, these methods do not consider the relationships between time segments during testing. In contrast, by inferring over a video-level graph, our method can use information from the entire video during training *as well as testing* to achieve better localization. Recent

work [148] uses recurrent neural networks to model relationships between time segments. However, relationships between time segments that are temporally distant, or that belong to different videos cannot be modeled. In contrast, our model is not restricted by temporal proximity when modeling similarity and dissimilarity relationships between time segments.

Some work use additional cues such as person detection [157, 158], scripts/subtitles [159, 160, 161], or external text [162]. Others use activity ordering information to assist in discriminative clustering [163, 164], temporal alignment [165, 166, 167], and segmenting temporal proposals [168, 169].

A growing body of work explore neural network based graphs [149, 170]. In computer vision, graph convolutions have gained popularity for capturing relationships between objects spatially and temporally for video object understanding, as well as capturing spatio-temporal dynamics for action understanding [171, 172, 173, 174, 175, 176, 177]. In particular, [171] develops an LSTM based graph for video object detection that uses strong action localization annotation as supervision. Unlike our method they do not use graph convolutions, and operate in a different 'slightly supervised' setting for video object detection, where human action labels are used to generate object detection labels. [173] uses both an appearance similarity graph alongside a temporal similarity graph to understand relations between video regions for action classification. However, unlike our method, it operates in a fully-supervised setting.

5.2 Approach

Our goal is to train a temporal action localization system that predicts the start and end times of each action's occurrence in a video. During training, we are only provided with weak action labels: we know what actions occur in a video but we do not know when or how many times they occur. We use these weak action label–video pairs to train our system. During testing, input videos have no labels.

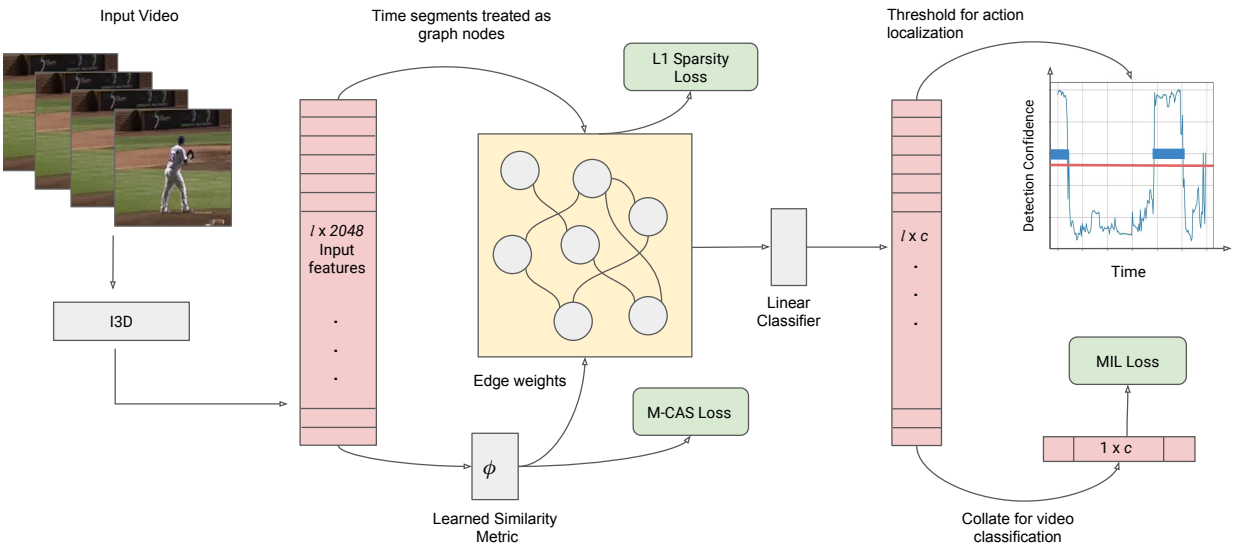

Figure 5.2: **Method Overview:** We use a pre-trained I3D network to extract input features for each time segment in a video. Each time segment is represented as a graph node, and edges between nodes are weighted by their level of learned similarity. Segment-level classification predictions are made by inference over this graph. During test time, we threshold the segment-level predictions to get activity localization predictions. We use a Multiple-Instance Learning (MIL) loss to supervise the classification, an L1 loss on edge weights to keep the edges in the graph sparse, and a modified Co-Activity Similarity Loss (M-CASL) to encourage edges between foreground segments to be higher than edges between foreground and background segments.

5.2.1 Architecture

Our network architecture is shown in Fig. 5.2. The input to our network is an $l \times d_{in}$ volume of features, where l is the number of input time segments in the video, and d_{in} is feature dimension. We refer to each time segment's input feature as $\mathbf{x}$ and the entire input volume as $\mathbf{X}$. The input features are then transformed using a graph convolution layer. We use RGB and optical flow based similarity to weight edges in the graph, where the similarity metric is learned by a separate linear layer ϕ. For each input time segment, the network outputs a prediction confidence for all classes. We refer to the final prediction $l \times c$ volume as $\mathbf{Y}$, where c is the number of action classes.

5.2.2 Feature extraction

We extract features from a Kinetics pre-trained I3D [178] to represent each video segment, as in [144]. Specifically, each video is represented by two $l \times 1024$ volumes (where l is the number of input time segments), one extracted from a RGB based stream and one extracted from an optical flow based stream. These volumes are concatenated to give a final $l \times 2048$ representation. Each time segment corresponds to 16 frames extracted at 25 FPS, or 0.64 seconds.

5.2.3 Graph convolution layer

Each input time segment is treated as a node in a graph over which inference is performed. The node edges are weighted by their similarity. In this way, related time segments can be pushed together and unrelated time segments can be pushed apart in feature space, while informing one another during both training and testing phases. Through this process, the graph convolutions can encourage better localization as the network is forced to inspect and predict each time segment class in the context of other time segments that it is similar to as well as different from.

The graph layer performs the following transformation on input $\mathbf{X}$:

$$\mathbf{Z} = \widehat{\mathbf{G}}\mathbf{X}\mathbf{W}$$

where $\mathbf{Z}$ is an $l \times d_{out}$ output of the graph convolution, $\mathbf{W}$ is a $2048 \times d_{out}$ weight matrix learned via backpropagation, and $\widehat{\mathbf{G}}$ is the row normalized affinity matrix $\mathbf{G}$. $\mathbf{G}$ is an $l \times l$ affinity matrix where $\mathbf{G}_{ij}$ is the edge weight between $\mathbf{x}_i$ and $\mathbf{x}_j$.

To compute $\mathbf{G}$, we first learn a simple affine function ϕ on input feature $\mathbf{x}$:

$$\phi(\mathbf{x}) = \mathbf{w}\mathbf{x} + \mathbf{b}$$

where $\mathbf{w}$ and $\mathbf{b}$ are weight and bias terms. ϕ is used to weight graph edges such that nodes with more similar ϕ have higher edge weights between them. $\mathbf{G}_{ij}$ (edge weight between $\mathbf{x_i}$ and $\mathbf{x_j}$) is computed as:

$$\mathbf{G}_{ij} = f(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j))$$

where $f(\cdot)$ is cosine similarity.

$\mathbf{G}$ essentially transforms each row of $\mathbf{X}$ to a weighted combination of other rows of $\mathbf{X}$. Note that this formulation subsumes other common layer operations. An identity $\mathbf{G}$ corresponds a regular fully connected layer with no bias term. A $\mathbf{G}$ with zero off-diagonal values, and non uniform diagonal entries works similarly to an attention mechanism. By setting rows of $\mathbf{G}$ to one or zero, average and max pooling operations can be performed. Multiple graph layers can be stacked together as the $\mathbf{Z}$ of the layer below becomes the $\mathbf{X}$ of the layer above. However, due to the small size of our datasets we use only a single graph layer. The output of our graph layer is passed to a linear classification layer to obtain the final $l \times c$ volume $\mathbf{Y}$.

5.2.4 Loss functions

Our method uses three separate losses. We use a multi-instance cross entropy loss that trains the network to correctly classify each video via segment level classification. We also impose an L1 sparsity loss on our graph so that graph edges are sparse and discriminative time segments can be clustered together. Last, we impose a co-activity similarity loss on the learned similarity function ϕ, so that salient parts for each video class are encouraged to have high edge weights between them.

Multiple instance learning loss

Similar to prior work [146, 144], we treat the problem of weak action localization as a multiple instance learning (MIL) problem. Each video is treated as a bag of instances, some of which

are positive instances. We only have video-level labels, and must use them to correctly classify instances within each video. To do this, we classify all instances, and then average the classification predictions for the top k per class to get a c dimension video-level prediction vector. The vector is normalized using softmax so that each dimension, p_i represents the probability for class i. At the same time, the binary indicator ground truth vector y (a video can contain multiple action classes) for a video is normalized so that it sums to 1. It is then used alongside the video prediction vector to calculate the multi-class cross entropy loss averaged across a batch of n videos, indexed by j:

$$L_{MIL} = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{C} -y_i^j \log p_i^j.$$

We set k to $\max(1, \lfloor \frac{l}{d} \rfloor)$ where l is the total number of input features for a video, and d is a hyper parameter. We further analyze the effect of d in Section 5.3.2. This part of our framework is similar to the multiple instance learning loss branch in [144] and the hard selection module of [146]. Unlike binary cross entropy loss, this loss formulation gives equal weight to each training video rather than each label occurrence. Hence, instances of each class that occur in videos with fewer labels get more weight than instances that co-occur with many other classes, which we found leads to better performance than the binary cross entropy loss.

Graph sparsity loss

To recap, $\mathbf{G}$ transforms rows of $\mathbf{X}$ to a weighted average of rows of $\mathbf{X}$. In other words, $\mathbf{G}$ can cluster together similar $\mathbf{x}$'s, and push apart dissimilar $\mathbf{x}$'s. However, a $\mathbf{G}$ with edge weights that are close to uniform will make it hard for the network to train, as discriminative signals in $\mathbf{X}$ will be averaged out. In order to prevent this, we enforce edge weights in $\mathbf{G}$ to be sparse by imposing an L1 loss on the absolute sum of $\mathbf{G}$:

$$L_{L1} = \frac{\sum_{i=1}^{l} \sum_{j=1}^{l} |\mathbf{G}_{ij}|}{l^2}$$

The loss works to encourage sparsity in $\mathbf{G}$, and hence trains ϕ to create tighter clusters from $\mathbf{X}$.

We find that it is helpful to additionally ignore edges that have a low absolute value. We therefore drop edges in each graph that are in the lower half of its range of edge weights.

Modified co-activity similarity loss

Our last loss is a modification of the Co-Activity Similarity Loss (CASL) [144]. It supervises the intermediate feature representation corresponding to video segments by both increasing the distance between foreground and background features, and decreasing the distance between foreground features of the same class.

The foreground and background representations are the sum of time segments' intermediate feature representations weighed by their predicted classification confidence. Specifically, for a given video, let $\mathbf{F}_t$ represent the intermediate feature representation of time segment t, let $p_{i,t}$ represent the classification confidence of time segment t belonging to class i, and let $\hat{p}_{i,t}$ represent $p_{i,t}$ after softmax normalization across all classes to segment t. The foreground feature representation $\mathbf{f}_i$, and background feature representation $\mathbf{b}_i$, are then calculated as:

$$\mathbf{f}_i = \sum_{t=1}^{l} \hat{p}_{i,t} \, \mathbf{F}_t, \quad \mathbf{b}_i = \sum_{t=1}^{l} (1 - \hat{p}_{i,t}) \, \mathbf{F}_t$$

where l is the total number of time segments in the video.

For a video j and its ground truth action class i, the foreground $\mathbf{f}_i^j$ and background $\mathbf{b}_i^j$ feature representations are obtained. For any two videos j and k belonging to the same class i, their foreground and background representations can then be used to calculate the

Co-Activity Similarity Loss:

$$L_{CASL}^{j,k,i} = \max(0, \bar{f}(\mathbf{f}_i^j, \mathbf{f}_i^k) - \bar{f}(\mathbf{b}_i^j, \mathbf{f}_i^k) + 0.5) + \max(0, \bar{f}(\mathbf{f}_i^j, \mathbf{f}_i^k) - \bar{f}(\mathbf{b}_i^k, \mathbf{f}_i^j) + 0.5)$$

where $\bar{f}(a, b)$ is cosine distance and 0.5 is the margin.

CASL was originally designed to supervise the intermediate feature representation that is used to make the final class wise predictions; i.e., an unmodified use of CASL would be on the output of our graph convolution layer. Here, we instead apply the loss on the output of ϕ. That is, we use CASL to encourage the edge weight between two foreground segments a and b of the same class to be high (and the edge weight of a foreground and background segment to be low). This affects how rows of $\mathbf{X}$ are averaged. It does not directly supervise the learned weight matrix $\mathbf{W}$; $\mathbf{W}$ is still free to transform rows a and b of $\mathbf{GX}$ differently. In this sense, our modified CASL (MCASL), i.e. applying CASL on ϕ, is a less rigid imposition of the loss, one that would not be possible in a regular fully connected layer. In Section 5.3.2, we show that this choice is more effective in reducing overfitting than directly supervising the intermediate feature representation.

Final loss

The final loss used to supervise the training is:

$$L_{Total} = \lambda_1 L_{MIL} + \lambda_2 L_{L1} + \lambda_3 L_{CASL}.$$

We set $\lambda_1 = \lambda_2 = \lambda_3 = 1$. These hyperparameters are set so that no one loss dominates training.

5.2.5 Action classification and localization

During test time, we input a single video, and obtain an $l \times c$ volume output $\mathbf{Y}$. We average the top k segments per class to obtain a video-level classification prediction.

In order to obtain hard localization predictions (video segment classifications), we threshold the confidence values to ignore the lowest 5% range of predictions. We merge temporally consecutive time segments that are classified as the same action into a single detection, and assign it the maximum confidence of its merged segments. We use these detections for the final evaluation.

5.3 Experiments

We evaluate our approach against state-of-the-art weakly-supervised temporal action localization methods. We also analyze the effects of edge sparsity and our different losses. Lastly, we present qualitative and quantitative results that highlight the advantage of our graph-based approach over traditional methods that do not explicitly model the relationship between time segments.

Datasets We present results on three datasets, of which THUMOS'14 and ActivityNet 1.2 have been previously used to evaluate weakly supervised action localization.

THUMOS'14 [150] has temporal annotations for 20 classes, with 200/211 untrimmed validation/test videos. Each video contains one or more of the 20 classes, with an average of 1.12 classes per video. We use the validation dataset for training, and the testing data for testing.

ActivityNet 1.2 [151] comprises 4819 training videos, 2383 validation videos, and 2480 test videos with withheld labels. There are a 100 action classes with an average of 1.5

temporal activity segments per video. We use the training videos as training data, and validation videos as test data.

Charades [152] is composed of 9848 videos, with 7985 as training videos, and 1863 as validation videos. The videos have an average length of just 30 seconds, and feature fine grained actions such as 'Putting Clothes Somewhere' and 'Throwing Clothes Somewhere' performed in visually similar indoor settings. Videos have an average of 6.75 actions. We use features extracted from i3D network finetuned on Charades [178].

Implementation details The output of ϕ as well as our graph layer is 1024. The output of the graph layer is passed through a ReLU non-linearity and then L2 normalized before being passed to the linear classification layer. We use Dropout at 0.5 between the graph and linear layer. The output of the classification layer is passed through a Tanh layer to obtain the final class confidence values. The final Tanh non-linearity limits the range of class confidence scores so that a standard threshold of -0.9 can be applied across all datasets. Using a standard threshold ensures that we do not trivially inflate performance for datasets with longer actions by predicting the full duration of each video.

Though not encountered in our experiments, the graph layer's matrix multiplication $\mathbf{GX}$ can run into GPU memory limitations for large graphs. During train time, the number of time segments per graph can be limited, and during test time $\mathbf{G}$ and $\mathbf{GX}$ can be calculated offline on CPU, or in smaller row wise chunks on GPU as a solution.

We train for 250 epochs with Adam [179] at a learning rate of 0.001. During both training and testing we build $\mathbf{G}$ from time segments from a single video at a time.

For THUMOS'14, we use a batch size of 32 videos and calculate the CAS loss for every pair of videos with the same ground truth class label. For the larger ActivityNet 1.2 and Charades, we use a batch size of 256. Since calculating the CAS loss for every pair of videos for this larger batch size increases the required training time exponentially, we fix half of each batch with video pairs that have a randomly picked class in common. The CAS loss is

Method	mAP@IoU		
	0.5	0.7	0.9
UntrimmedNets [146]	7.4	3.9	1.2
Auto-Loc [155]	27.3	17.5	6.8
W-TALC [144]	**37.0**	14.6	-
Ours	29.4	**17.5**	**7.5**

Method	mAP
Sigurdsson et al. [152]	12.8
SSN [180]	16.4
Super Events [181]	19.4
TGM [182]	**22.3**
Ours	**15.8**

Figure 5.3: **(Left)** Localization performance on ActivityNet 1.2 val set. **(Right)** Localization performance on Charades. All methods except 'Ours' are strongly supervised

	mAP@IoU					
Method	0.1	0.2	0.3	0.4	0.5	Cls
HAS [153]	36.4	27.8	19.5	12.7	6.8	-
UntrimmedNets [146]	44.4	37.7	28.2	21.1	13.7	74.2
STPN (UNTF) [145]	45.3	38.8	31.1	23.5	16.2	-
STPN (I3DF) [145]	52.0	44.7	35.5	25.8	16.9	-
AutoLoc [155]	-	-	35.8	29.0	21.2	-
W-TALC (UNTF) [144]	49.0	42.8	32.0	26.0	18.8	-
W-TALC (I3DF) [144]	55.2	49.6	40.1	31.1	22.8	85.6
MAAN [143]	59.8	50.8	41.1	30.6	20.3	94.1
Ours	**63.7**	**56.9**	**47.3**	**36.4**	**26.1**	**94.2**
STAR* [148]	**68.8**	**60.0**	**48.7**	34.7	23.0	-
Ours	63.7	56.9	47.3	**36.4**	**26.1**	**94.2**

Table 5.1: Localization performance on Thumos'14 test set. The last column shows video classification performance. Asterisk indicates the method uses additional annotation.

then only calculated for the paired videos.

5.3.1 Comparison to state-of-the-art

Table 5.1 and Figure 5.3 (left) show weakly-supervised temporal action localization results on THUMOS'14 and Activity 1.2, respectively. We use mean average precision (mAP) to calculate localization accuracy at different overlap thresholds. Overlap threshold is used to determine the minimum required overlap between a ground truth occurrence and a prediction for it to count as a true positive.

For THUMOS'14, our method outperforms all previous methods at the challenging overlap threshold of 0.5, with a margin of more than 3 mAP points. This gap in performance is

	mAP@IoU				
Method	0.1	0.2	0.3	0.4	0.5
Baseline	26.1	19.4	13.1	8.9	5.8
MCASL	26.7	20.8	14.7	9.9	6.2
L1	55.3	46.9	39.0	28.5	19.6
L1+MCASL	**63.7**	**56.9**	**47.3**	**36.4**	**26.1**

Table 5.2: Ablation study of different constraints on our appearance similarity graph on Thumos '14 test set.

retained even when comparing against STAR [148] which uses additional annotation in the form of the number of times an action occurs in a video during training. Similarly, we outperform previous methods on ActivityNet 1.2 at higher overlap thresholds. To demonstrate localization ability independent of classification, we also calculate mAP for ground truth action classes. This results in 19.7% and 8.2% mAP at 0.7 and 0.9 IoU for ActivityNet, and a slight increase at 0.5 IoU to 63.9% for THUMOS'14.

Figure 5.3 (right) shows additional results of our method on Charades. While our method is 6.5 points below the state-of-the-art in a fully supervised setting, it is 3 points higher than its original fully supervised baseline and presents a challenging weakly supervised baseline for future methods to compare with. Like previous methods, we report mAP for 25 equally spaced time points in each video.

5.3.2 Ablation studies

We next study the effect of our three losses. In particular, we study the effect of CASL by showing that it is more effective with a graph-based method than an approach that does not explicitly cluster time segments together. We show that the modified CASL is able to do better by guarding against over-fitting. Last, we inspect how k should be set for the top k multi instance learning loss.

Graph supervision

We first analyze the importance of each constraint on the appearance similarity graph. The appearance similarity graph uses an L1 loss to encourage non-uniform edge weights, and

	mAP@IoU				
Method	0.1	0.2	0.3	0.4	0.5
FC-CASL 1024	55.1	47.9	38.4	29.4	18.3
FC-CASL 2048	55.4	48.3	40.0	30.3	19.8
CASL-Graph	57.7	50.9	42.0	32.1	22.5
MCASL (Ours)	**63.7**	**56.9**	**47.3**	**36.4**	**26.1**

Table 5.3: Using a graph with CASL (last two rows) is more effective than using regular linear layers (FC-CASL rows) since it explicitly utilizes relationships between temporal segments.

a co-activity similarity loss (CASL) on ϕ to supervise edge clustering. Table 5.2 shows the results of our ablation study. L1 loss is most crucial for performance, as it more than triple the performance at 0.5 overlap. MCASL provides the next significant improvement: a 8.4 mAP improvement at 0.1 IoU threshold. While MCASL improves performance of the baseline model, it is most useful when working with an L1 loss. This indicates MCASL is more useful when working with a sparse graph.

Modified co-activity similarity loss

We next analyze the effect of the co-activity similarity loss.

We develop a baseline model that uses the CASL loss without a graph convolution layer to contrast it with our graph-based approach. Specifically, the model uses a fully-connected layer instead of a graph layer, but is otherwise identical. The resulting model 'FC-CASL 1024' has a 1024 dimension intermediate output like our graph model. We also train a higher-capacity model 'FC-CASL 2048' that has a larger intermediate layer with a 2048 dimension output, which is roughly the same number of learnable parameters as ours. These baseline models are very similar to the model in [144], except they have the same non-linearities as our network. These are also equivalent to our network without a learned similarity metric ϕ, but a fixed identity adjacency matrix $\mathbf{G}$. We additionally develop a baseline model that uses the original CAS loss 'CASL-Graph': instead of applying CASL on the output of ϕ as done in our model, we apply it to the output of our graph layer. Thus, the only difference between this baseline and the 'FC-CASL' baselines is the graph layer.

Table 5.3 shows the results. Applying the CASL loss on the output of the graph layer

d	Video %	THUMOS mAP @ 0.5 IoU	THUMOS Test Data %	ActivityNet mAP @ 0.5 IoU	ActivityNet Test Data %	Charades mAP Per Frame	Charades Test Data %
1	50-100%	18.5	2.8	**29.4**	57.2	14.9	76.9
2	25-50%	44.9	3.8	5.5	19.0	15.4	82.0
4	12.5-25%	58.4	14.1	1.7	14.4	15.2	75.5
8	0-12.5%	**63.7**	93.9	1.4	18.8	13.8	15.4
Random		39.0	-	14.3	-	**15.8**	-

Table 5.4: Setting hyperparameter d to correspond with expected action duration results in the best performance across datasets.

'CASL-Graph', leads to a ∼3 mAP improvement over the 'FC-CASL' baselines. This points to the superiority of using a graph based approach for weakly-supervised action localization versus relying on conventional linear layers. In addition, the better performance of our full model compared to 'CASL-Graph' shows that our modified CASL which supervises input feature clustering, rather than intermediate network features is a better method for providing supervision. By tracking testing performance throughout training, we find that 'CASL-Graph' begins to overfit midway through training after reaching peak performance at 59 mAP at 0.1 IoU. On the other hand, 'Ours-MCASL' reaches higher peak performance and then maintains it through the end of training since it is not modifying the actual intermediate feature representation of the network, but only modifies how the input I3D features are clustered for further inference.

MIL Loss Parameter

As explained in Section 5.2.4, the multi instance learning loss is calculated over the average of the top k predictions of each class. k is chosen to be 1/8 of the length of a video by setting parameter d. While $d = 8$ works well for THUMOS'14, it is not optimal for ActivityNet and Charades.

Generally speaking a smaller d (or larger k) results in longer detections as the MIL loss is backpropagated to more time segments at every iteration. Table 5.4 shows the performance of our system for different values of d against the percent of test videos that feature activities

Figure 5.4: **Visualizing graphs:** Strongly connected graph cliques are shown in blue. In red we show segments that are considered very dissimilar to the foreground segments. The corresponding adjacency matrix for each example is shown on the right.

with corresponding duration. The d that results in the best performance mimics the activity duration bias for each dataset; 57% of ActivityNet test videos feature actions that last more than half the video length, so setting $d = 1$ during training results in the best performance. With very short action durations, THUMOS'14 performs best with a large d or shorter predictions. Without prior knowledge of typical activity duration, or a temporally labeled validation set that can be used to set d, one useful strategy is to randomly choose a value for d for each training iteration. The last row shows results where d is randomly selected from the set $\{1, 2, 4, 8\}$ every training iteration. With a balanced activity duration, 'Random' is the best strategy for Charades, and for both ActivityNet and THUMOS'14 results in performance that is significantly better than the worst d setting, but about half of the optimal level. Estimating d without any temporal annotation is an interesting direction for future research.

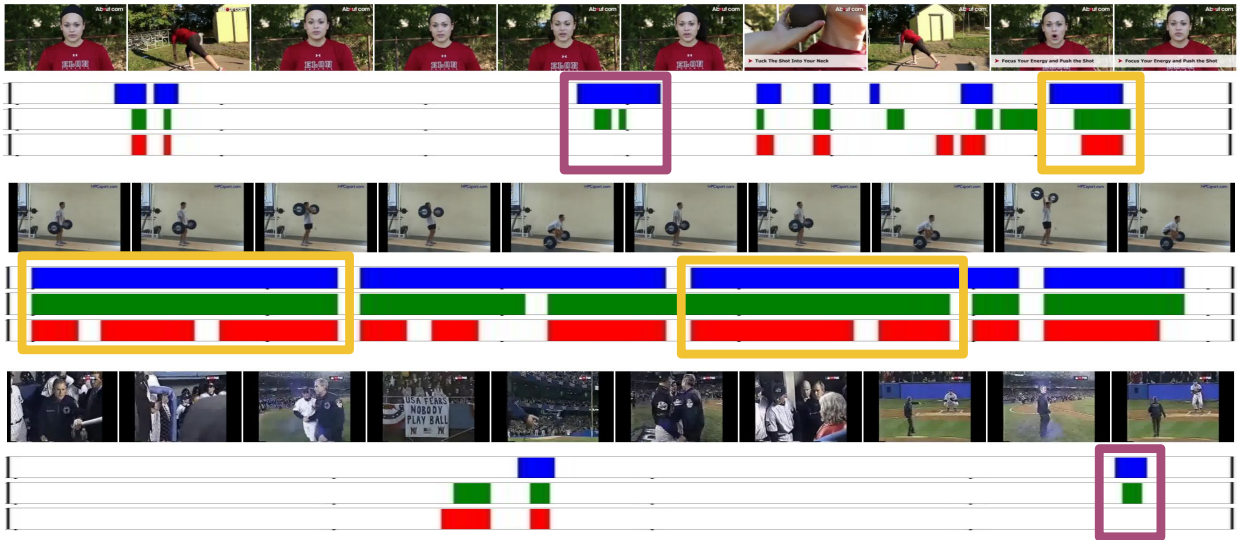

Figure 5.5: **Qualitative Comparison:** The ground truth is in blue, our detections are in green, and a baseline without a graph ('FC-CASL') results are in red. The video frames are sampled uniformly across the video length. By using similarity across time segments to make our predictions, our method is able to localize larger extents of actions (yellow) and is able to develop a more general model of action classes that allows it to localize to more instances of an action (magenta).

5.3.3 Qualitative results

Figure 5.5 shows some qualitative results comparing our method against baselines. Ground truth, our results, and the 'FC-CASL' baseline results are shown in blue, green and red, respectively, for videos from different classes. Using a graph allows our network to localize actions with more overlap (in yellow). This is most apparent in the second row, where our detections are not split up and wider than the baseline's. Our model is also able to localize more occurrences of different actions; in magenta we show instances that are not detected by 'FC-CASL' but are detected by our method.

Figure 5.6 shows additional qualitative results from the THUMOS '14 dataset. The ground truth is shown in blue, with our detections in green. Overall, our method is good at localizing all occurrences of an action.

The first row shows an example of a video with multiple 'Hammer Throw' occurrences during most of the video, followed by a few occurrences of 'Clean and Jerk'. Our method is

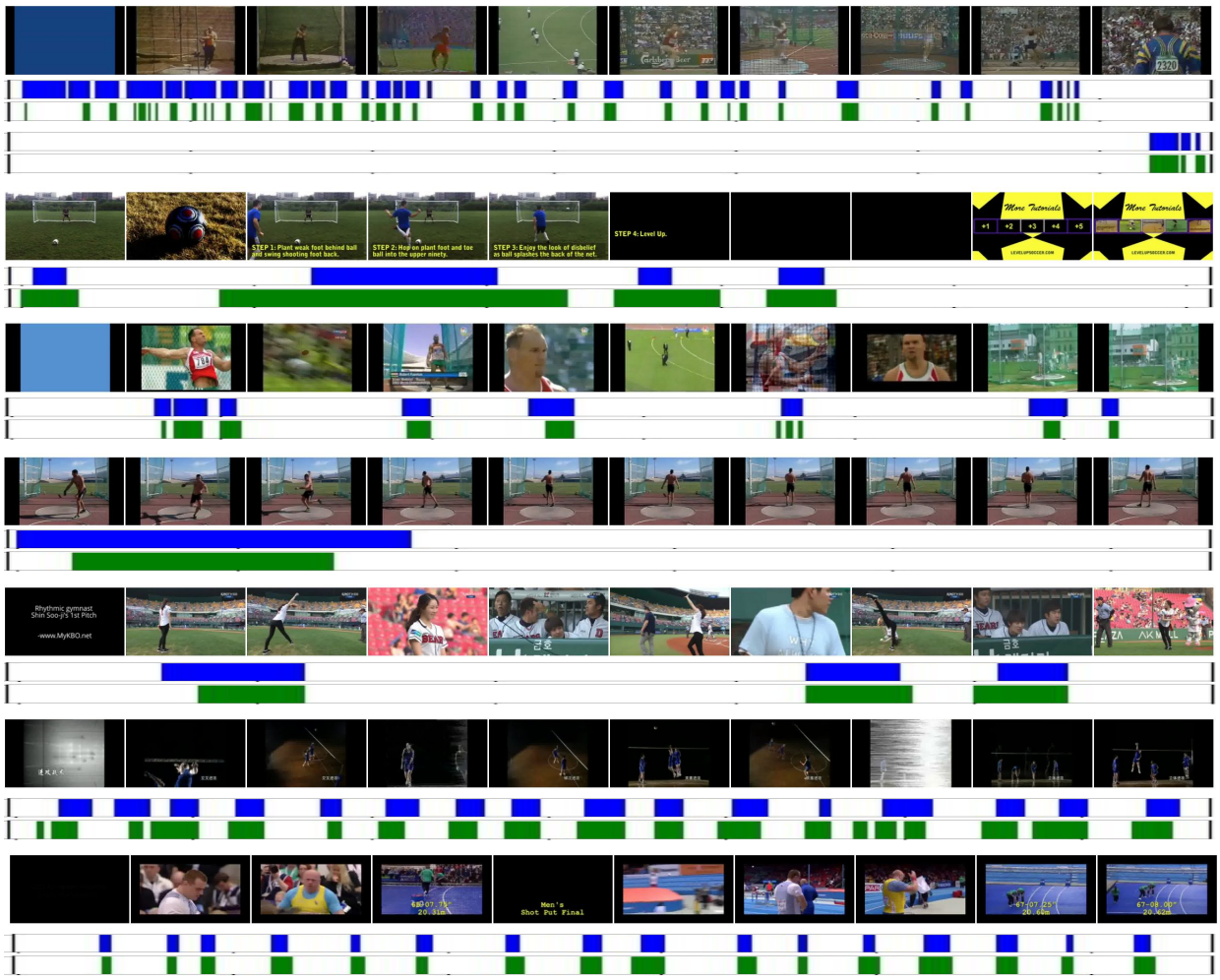

Figure 5.6: **Qualitative results:** The groundtruth is in blue, and our detections are in green.

able to localize almost all occurrences, however sometimes the localizations are too short in length, or broken in to multiple occurrences. On the other hand, in the second example of 'Soccer Penalty', our model provides localizations that are a little too long compared to the ground truth.

In Figure 5.7 we show some failure examples of our system. Multiple action occurrences that happen close in time are lumped together as a single detection in the first and second examples for the actions of 'Tennis Swing' and 'Cricket Bowling'. However, the network is able to distinguish multiple occurrences of both actions from longer segments of time when no action is happening, as indicated by the lack of false positives. While our network is able

Figure 5.7: **Failure results:** The groundtruth is in blue, and our detections are in green.

to localize almost all instances of 'Shot Put' in the third row, our detections do not span the full duration of the action and have poor overlap. Finally, our network fails completely in the last example of 'Volleyball Spiking', where it localizes the start and end of the spiking action rather than its actual duration.

5.3.4 Visualizing Graphs

In Figure 5.4 we show the adjacency matrix of two graphs, and the nodes that form high edge cliques in these graphs. All images are *not* temporally neighboring, and taken from different points in the video. Graph cliques are surrounded in a blue box. Segments that are considered dissimilar to the foreground segments are surrounded in a red box.

In the cricket bowling video, the graph forms cliques from parts of cricket bowling so that the start of the ball throw forms one cluster, the arm swing forms another, and so on. The segments considered least similar to bowling segments are shown in red and show batting, and a zoomed out view of the stadium; segments with very little relevance to the bowling action.

The second example shows a video with three distinct cliques. The video features a man explaining how to score a soccer penalty, and then demonstrating it repeatedly. The largest clique lumps together nodes where the man is facing the camera and talking. Another clique

comprises the action right before the soccer penalty – placing the ball and taking the starting position. The last clique lumps together the actual soccer penalty.

These examples show some interesting ways the graph can cluster nodes – it can cluster together subactions of an action class, and structured activities that may be relevant to, but distinct from, the action class.

5.4 Discussion

We presented a novel approach for weakly supervised temporal action localization. Without frame level annotation during training, an action localization system must necessarily infer action categories from the similarity and difference between time segments of videos. Despite this, current methods do not make explicit use of appearance and motion similarity between time segments to inform predictions. In contrast, our method makes explicit use of similarity relationships between time segments by using graph convolutions. As a result, it is able to harness similarity relationships to develop a better model of each action category, and is consequently able to localize actions to a fuller extent. We pushed the state of the art on Thumos'14 and ActivityNet 1.2 for weakly supervised action localization, and presented the first results on Charades. We demonstrated quantitatively and qualitatively that a baseline approach that does not use graph similarity achieves inferior performance. Last, we demonstrated through ablation studies the importance of each component of our system, and presented analysis of the weaknesses of our approach.

Chapter 6

Equine pain behaviour detection via self-supervised disentangled latent pose representation

The last chapter presented a weakly supervised method for action localization. In this chapter, we apply weakly supervised detection explicitly for the task of equine pain detection.

Equine pain detection is a challenging problem, with expert human performance on video data at just 58% accuracy for pain or no pain classification [28]. While self evaluation can be used as a gold standard for determining pain in human subjects, horses being non-verbal lack a gold standard for pain [183]. In addition, as prey animals horses hide signs of pain from humans [19]. It is therefore difficult to ascertain if a horse is experiencing and expressing pain.

Determining the visual signs of pain in horses is an active area of research, and a variety of proxies for pain have been used for pain data labeling depending on the experiment design. For example, post operative horses are labeled as painful in [16] while Gleerup et al [17] treats time periods when pneumatic pressure, and capsaicin cream are applied to horses as periods of induced pain. At the same time, overlays of different emotions such as drowsiness

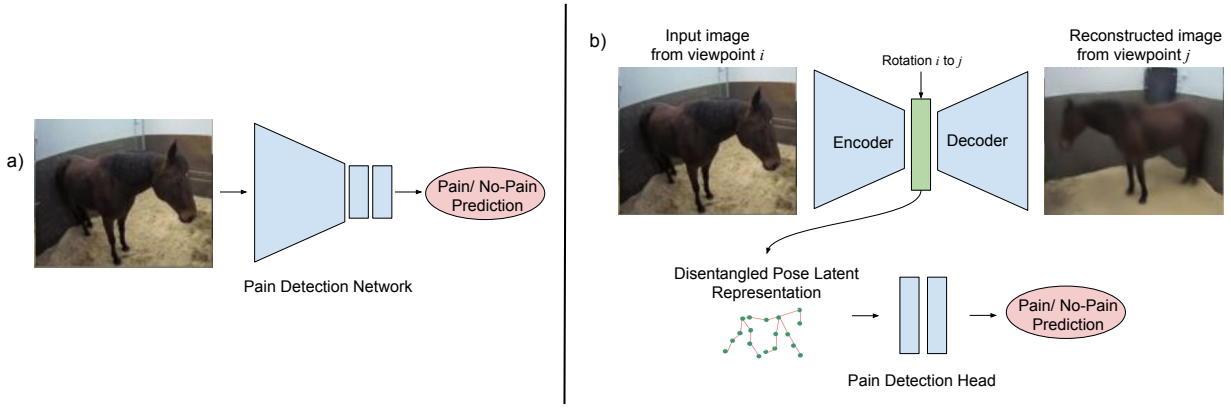

Figure 6.1: **Main Idea** (a) Directly training a pain classifier on video frames can result in a model that is not interpretable, and overfitted to training data. Our two step approach (b) first uses self supervised multi-view synthesis to learn a latent horse pose representation, and then second uses the disentangled pose representation to learn a light weight pain classifier.

or stress may also exist and further complicate deducing pain expressions in horses.

In veterinary practice pain scales that use facial and body behaviors in combination with records of time spent performing different activities such as eating – activity budgets – are used to determine the pain level of horses [7, 8, 9, 10, 12]. However, outside of an animal hospital or clinic, determining horse pain can be very difficult as it requires both frequent observations and expert evaluation. An expert trained computer vision system capable of determining horse pain, therefore, has great potential for improving animal welfare by timely detection of pain and consequent diagnosis of illness.

Datasets with detailed annotation have been used for training human pain detection systems [184]. However, a similar model of pain detection may not be practical for equine pain detection. Obtaining similar detailed annotation is very costly and time consuming. For example, it takes an Equine Facial Action Coding System [43] annotator upwards of 30 minutes to label a 1 minute clip [185]. In addition, with pain scales including entries like 'interactive behavior' there is not always a mapping between pain attribute and obvious visual behavior [12]. Finally, horses behave differently when aware of being observed than when there are no humans present [19], which calls to question the applicability of datasets with observed horses to more natural settings when the horse is alone. Relatedly, a vision based

monitoring system for horses would be more pragmatic if it could operate off unobtrusive surveillance cameras, rather than require horses to be close to the camera, with the face clearly visible as is true for human datasets.

In 2018-2019, a dataset of horses with induced orthopaedic pain was collected at Swedish University of Agricultural Sciences [48]. It included surveillance video footage along with pain labels from periodic pain assessments by expert veterinary researchers. This work uses the surveillance video data, along with coarse pain labels to determine the pain of unobserved horses. While, the dataset is a rich source of unobserved horse data, the dataset contains few subjects (8), and lacks detailed pain annotation. A fully supervised feed forward pain detection network is therefore likely to overfit to the training data. At the same time, the network predictions would not be interpretable, and may use extraneous information, such as the time of the day, or the lighting in the stall to determine the pain state of the horse.

On the other hand, self supervised methods have been shown to disentangle semantically and visually meaningful properties from image data without the use of labels. Examples include disentangling the 3D normals, albedo, lighting, and alpha matte for faces [186], and disentangling pose and identity for facial recognition [187].

Our **key idea** is to use self supervision to disentangle the visual properties we would like a pain detection system to focus on, and then use the disentangled representation for identifying pain. In this manner we can reduce the likelihood of the model learning extraneous information to determine pain, and prevent overfitting to the training data (see Figure 6.1).

We use a two step process for pain detection. The first stage we train a view synthesis model that, given a frame of a horse from one angle learns to synthesize the scene from a different viewpoint [188]. We use an encoder-decoder architecture, and disentangle the horse pose, identity, and background in the process. In the second stage, we use the learned pose representation to classify video segments as painful. As we lack detailed temporal annotation for pain, we use weak supervision to train the pain classification module, and propose a modified multi-instance learning loss towards this end. Our system is able to learn

a viewpoint aware latent pose representation, and determine the pain label of video segments with 60% accuracy. In comparison, human performance on a more close-up dataset of horse facial videos was at 58% accuracy [28].

We present pain detection results of our model with ablation studies comparing the contribution of each module. In addition, we visualize and analyze the features of pain detected by our system, and note their correspondence with current veterinary knowledge on equine pain behavior. Our contributions are:

- Creating a disentangled horse pose representation using a self supervised novel view synthesis method using surveillance video footage of horses in box stalls.
- Presenting a method for video level pain detection from the learned disentangled horse pose representation that is trained using weak pain labels and a novel modified multi instance learning loss.
- Extensive experiments including visualization and analysis of our automatically detected pain video segments and cues used for pain diagnosis in veterinary practice.

6.1 Related Work

Our work is closely related to the task of novel view synthesis, which is, given a view of a scene, generating images of the scene from new viewpoints. The task is challenging as it requires reasoning about the 3D structure and semantics of the scene from the input image. Rhodin et al [188] make use of synchronized multi-view data to create a disentangled pose, and identity latent representation by training an auto-encoder type architecture to use input images from one view to synthesize images of the same time instant from a different view point. Our work uses the same approach to learn a disentangled pose representation. However, while their method uses the learned pose representation for the downstream and strongly related task of 3D and 2D body keypoint estimation, our work uses the latent representation to detect animal pain in a weakly supervised setting.

Other works achieve novel view synthesis by assistance from either noisy and incomplete [189, 190, 191, 192], or ground truth depth maps in addition to images during training [193, 194, 195, 196].

Similar to our work, generative models have been used with emphasis on learning a 3D aware latent representation. Of note is deep voxels [197], a persistent 3D feature volume designed to learn 3D scene representation when fused with lifted 3D object features, which has shown impressive results on synthetic data. Similarly, HoloGAN [198], uses real data and strong 3D priors during training to learn disentangled 3D pose, shape, and appearance features. Most recently, Synsin [199] proposes the use of point clouds and in-painting to transform and project latent 3D representations to novel viewpoints, and works with real rather than synthetic data. While the aim of creating a 3D aware latent representation is common between our work and these works, these methods emphasize the generation of accurate and realistic synthetic views, while our work uses the 3D representation for the downstream task of pain classification. While we do make use of multi-view data, the different viewpoints are few – 4 – and are separated by a wide baseline, unlike the above mentioned novel view synthesis works.

Generative models with disentangled latent representations have been developed for a wide range of purposes such as to discover intrinsic object properties like normals, albedo, lighting, and alpha matte for faces [186], fine grain class attributes for object discovery and clustering [200], and pose invariant identity features for face recognition [187], and have been a topic of extensive research [201, 202, 203, 204, 205]. Our work relies on a disentangled pose representation from multi-view data, and places emphasis on utilizing the learned representation for a downstream task. While self supervised disentangled pose representations have been used for the task of 2D and 3D keypoint recognition [188, 206, 207, 208, 209], no previous work has used self supervised disentangled pose representations for the behavior related task of pain recognition, particularly in animals.

There is a growing body of work on deep visual learning for animal behavior and body

understanding. This includes work on animal body keypoint prediction [210, 211], facial keypoint prediction [212, 79, 213], and dense 3D pose estimation via transfer from human datasets [214] and fitting a known 3D model to 2D image [215, 216]. Of note is [217] which uses a synthesized zebra dataset to train a network for predicting zebra pose, shape, and camera parameters at test time towards the ultimate goal of fitting a dense 3D model to a 2D input image. However, the method requires extensive 2D keypoint and segmentation annotation, and uses a known quadruped 3D shape basis [215] for the 3D model generation.

Beyond animal keypoint and pose prediction, there is a growing body of research on detecting animal emotional state from images and videos. Most relevant is Broomé et al's [28] work on horse pain recognition that uses a fully recurrent network for pain prediction on horse videos. Sheep [25], donkey [27], and mouse [26] pain have also been explored with promising results. At the intersection of body behavior and pain recognition lies [218], where 3D models are fit to 2D video data for horse lameness detection. However, previous methods use either facial data, strong supervision, or additional information such as keypoints, segmentation masks, or facial movement annotation to learn the pain models. On the other hand our work uses weak supervision, with no additional annotation, and uses video data with the full body of the horse visible rather than just the face.

6.2 Approach

We use a two step approach for pain detection. Our dataset comprises of time aligned videos of horses from multiple views. The data has coarse video level pain labels. In the first stage, we train an encoder-decoder type architecture for novel view synthesis, and learn an identity and viewpoint co-variant horse pose aware latent representation in the process. In the second stage, we train a pain detection head using the trained pose aware latent representation as input to diagnose pain in video sequences. Since the dataset does not have detailed temporal annotation of horse pain expression, we use a multi-instance learning

approach for pain detection for video sequences. In the following sections we first describe the LPS dataset, followed by details of our view synthesis, and pain detection methods.

6.2.1 Dataset

The LPS Dataset [48] comprises of 24 hour surveillance camera footage of 8 horses filmed before and during joint pain induction. The experimental protocol was approved by the Swedish Ethics Committee in accordance with the Swedish legislation on animal experiments (diary number 5.8.18-09822/2018). As few horses as possible were included and a fully reversible lameness induction model was used. Pain was induced by injecting a solution of lipopolysaccharide (LPS) in one of the horses' leg joints leading to swelling. The joint swelling may be painful, and makes it difficult for the horse to bear weight on the leg with induced swelling. The horses are stalled individually in one of two identical box stalls, with four surveillance cameras in each stall capturing round the clock footage of the horse behavior. Starting 1.5 hours after joint injection, horses were periodically removed from the stall for trotting and asymmetry movement measurements by expert veterinary researchers. Measurements were discontinued once horse movement asymmetry returned to baseline (pre-induction) measurements. In addition pain assessments were performed by direct observation 20 minutes before and after each movement asymmetry measurement.

Data Preprocessing

Of the larger dataset of collected over 16 days, we use video data from two hours of pre-induction baseline, and two hours of peak pain video footage for each horse from each surveillance camera. This results in 128 hours of data. We only use the time periods when no humans are present in the stall or the corridor outside the stall. This reduces the likelihood of curiosity or interactivity with external environment leading to confusing changes in horses' behavior.

Videos from each camera were manually offset when necessary to sync temporally with

other cameras in the stall. Videos belonging to the two hour peak pain period were labeled as pain videos, and pre-induction period were labeled as no-pain videos.

The intrinsic parameters for each camera are recovered by photographing a checkerboard pattern with known dimensions and solving Perspective-and-Point (PnP) problem using RANSAC in OpenCV [219].

With a large baseline between cameras it was not possible to capture the checkerboard in all cameras and recover cameras' extrinsic parameters with the same world coordinate system. However, for a subset of calibration instances the checkerboard was visible in three of the four cameras, with one camera in common between all calibration instances for each stall. Consequently, a common world coordinate system was determined by setting the global origin to the common camera's origin for all calibration instances. Following, we refined the intrinsic and extrinsic parameters of the cameras using bundle adjustment, i.e. by finding the camera parameters that minimize the reprojection error between the checkerboard corners in each image, and the projections of their corresponding 3D coordinates using the recovered camera parameters.

6.2.2 Multiview synthesis

Our multiview synthesis network uses the same architecture and training methodology as original work by Rhodin et al [188].

The model is a U-Net type architecture [220] that learns a disentangled horse identity and pose representation in its bottleneck layer. This is done by training the model to synthesize an image of an input image scene from a different viewpoint.

Specifically, given an input video frame $x_{v^i,t}$, from viewpoint i, at time t, the encoder, f_E, output is a latent representation of the horse pose, $p_{v^i,t}$, and a latent representation of the horse identity, $h_{v^i,t}$:

$$p_{v^i,t},\, h_{v^i,t} \;=\; f_E\left(x_{v^i,t}\right)$$

During training, both the pose and identity representations are manipulated before being inputted to the decoder. The pose representation is rotated by the relative camera rotations between camera viewpoint i and j, so that the pose representation input to the decoder, f_D, is in the same space as pose representations for viewpoint j:

$$p_{v^i \to v^j, t} \;=\; \mathbf{R}_{v^i \to v^j}\, p_{v^i, t}$$

The identity representation is swapped by the identity representation of an input frame of the same horse from the same viewpoint, but from a different time, t', and hence likely with a different pose. This identity representation swap encourages the network to disentangle pose and identity. In addition, a background image for each viewpoint, b_{v^i}, is input to the decoder so that the network does not focus on learning background information for synthesis and instead focuses on the horse:

$$x'_{v^i \to v^j, t} \;=\; f_D\left(p_{v^i \to v^j, t},\, h_{v^i, t'},\, b_{v^i}\right)$$

The synthesized image, $x'_{v^i \to v^j, t}$, is supervised by both a pixel-level mean square error loss compared with the ground truth image at time t from viewpoint j, as well as a perceptual loss on ImageNet pretrained ResNet18 [221] penultimate layers' feature.

$$L_{MVS} = ||x'_{v^i \to v^j, t} \;-\; x_{v^j, t}||^2 \;+\; \alpha\, ||\theta_{RN}(x'_{v^i \to v^j, t}) \;-\; \theta_{RN}(x_{v^j, t})||^2$$

where α is a loss weighting parameter, and function $\theta_{RN}(x)$ outputs ResNet18's penultimate feature representation for input image x. The multi-view synthesis loss, L_{MVS}, is averaged across all instances in a batch before backward propagation.

During testing, the synthesized image is generated without swapping of the identity representation.

Similar to [188], we detect and crop the horse in each frame to factor out scale and global

position. The rotation, **R** between two views is calculated with respect to the crop center instead of the image center and the crop is sheared so that it appears as if it were taken from a virtual camera pointing in the crop direction. In more detail, the crop is transformed by the homography induced by rotating the camera so that the ray through its origin aligns with the crop center.

Refining Multiview Synthesis For Horses

Unlike the Human 3.6 dataset [222] used in [188] features actors that move around constantly, the LPS dataset used by us features the horse standing or grazing in similar pose for long periods of time. As a result, randomly selecting a frame for the identity feature swap when training with LPS dataset can lead to suboptimal identity disentanglement. We therefore train on time sequences with a variety of horse poses – i.e. sequences with large optical flow – to achieve good identity disentanglement.

The background image for each view were at first extracted by taking the median images over all video frames from that view. However, the LPS dataset was collected over multiple months, during which the cameras were nudged by chewing from curious horses. As a result we calculated and used separate background images for each month. This lead to better background disentanglement.

6.2.3 Detecting Pain

The self supervised base network from Section 6.2.2 provides us with a means to disentangle the horse pose from its background, and identity from a given input image. The disentangled pose representation is then further used to train a pain detection head.

We perform pain detection with both frame and, following insight from previous research [28, 223] that showed pain detection from instantaneous observations to be unreliable, video clip level inputs. For clip level detection, we concatenate per frame pose latent representations in to a clip level volume that is then used as the atomic unit for pain detection

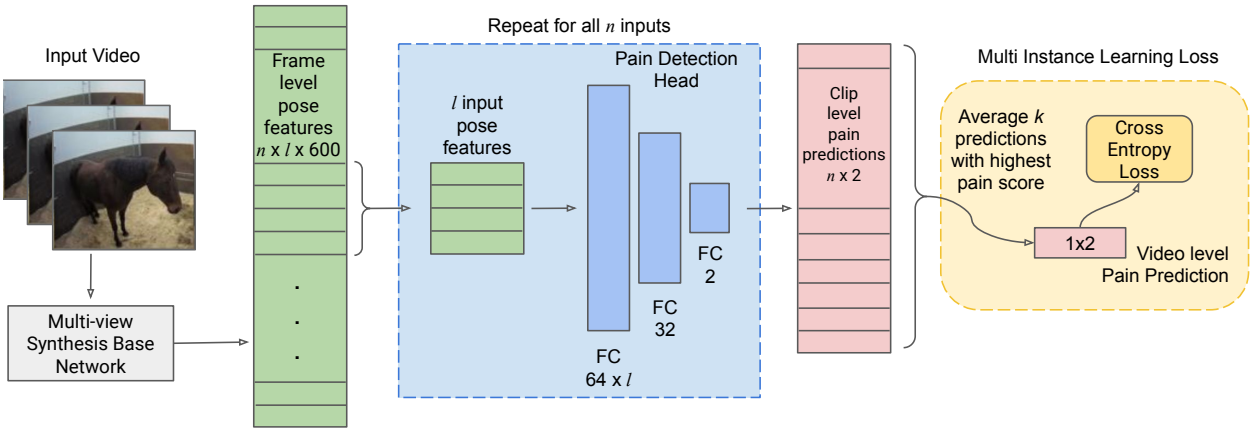

Figure 6.2: **Pain Detection Model.** Pose representations from the trained multi-view synthesis model are extracted for each frame for an input video, and collated into l length clips. The pain detection head, comprised of three fully connected (FC) layers, predicts pain for each clip, and clip level predictions are collated and supervised by the proposed multi instance learning loss.

during both training and testing.

The network architecture is shown in Figure 6.2 and comprises of two hidden linear layer with ReLU non-linearity and Dropout, followed by a classification layer with two dimensional output. Frame level predictions from the first linear layer are concatenated in to a $64 * l$ vector, where l is the number of frames in each time segment, that is then further forwarded through the network.

We use a multi instance learning setting for pain detection. Every video comprises of multiple time segments – either frames or clips – that are independently classified as pain or no pain segments. These segment level predictions are collated to obtain a video level pain prediction.

More specifically, each video sequence s, comprises of n time segments indexed by t. The pain head θ provides a two dimensional output that is softmaxed to obtain the pain, and no pain confidence values for time segment t, where boldface $\mathbf{p}$ represents the set of pose representations of all frames in time segment t:

$$y^{NP}_{v^i,t}\,,\; y^{P}_{v^i,t} \;=\; \text{softmax}\left(\theta\left(\mathbf{p}_{v^i,t}\right)\right)$$

The k time segments with the highest *pain* predictions are averaged to obtain the video level pain prediction:

$$y^P_{v^i,s} = \frac{1}{k}\sum_{t\in S} y^P_{v^i,t} \ , \ y^{NP}_{v^i,s} = \frac{1}{k}\sum_{t\in S} y^{NP}_{v^i,t}$$

where S is the set of k time segments' indices with the highest pain prediction:

$$S = \{j \,|\, y^P_{v^i,j} \in \max_K \{y^P_{v^i,1}, y^P_{v^i,2}, \ldots, y^P_{v^i,n}\}\}$$

The video level pain predictions are then supervised with a cross-entropy loss.

Selecting the top k segments with the highest pain predictions is an important modification to the multi instance learning loss used in literature (e.g. [146, 224, 225]), that would have averaged the highest k predictions for both pain and no-pain class independently to obtain the video level predictions. By collating only the predictions for the top k pain time segments to obtain the video level prediction, we do not penalize the network for detecting no-pain time segments within a pain video, and require only that the pain predictions have high confidence for a pain video, and no time segments have high pain confidence for a no-pain video. In Section 6.3.3 we show results without this loss modification.

Parameter k is set to be $\lfloor \frac{n}{d} \rfloor$ where d is randomly selected from the set $1, 2, 4, 8$ on every training iteration, and set to 8 during testing. Parameter d correlates with the proportion of a video that is predicted as an action class. As we do not know what proportion of time a horse in pain will express pain in a pain video, varying this parameter randomly is likely to provide the most robust performance as shown in previous work [226].

6.3 Experiments

6.3.1 Implementation Details

We use four hours of video footage per horse – two hours from before LPS injection, and two hours from the time period with maximum diagnosed pain, excluding time periods with

humans present in the stall.

MaskRCNN [227] is used to detect the horse in each frame. We noticed high confusion between 'horse' and 'cow' categories, and included high confidence detections from both categories. Detections were used to crop and center the horse in each frame.

Optical flow was calculated using Farnebäck's method [228] on video frames extracted at 10 frames per second. Time segments that had optical flow magnitude in the top 1%, 143559 frames, were used to train the multi-view synthesis module. We use leave one out subject exclusive training. Networks are trained for 50 epochs at 0.001 learning rate using Adam optimizer. The perceptual loss is weighted 2 times higher than the mean squared error loss during training. The same U-Net based architecture as in [188] is used. The dimension of the pose representation is 600, which is reshaped to 200×3 for multiplication with the viewpoint to viewpoint rotation transformation $\mathbf{R}$.

The pain detection dataset comprises of video segments with the maximum length of 2 minutes. Missing MaskRCNN detections can result in video segments of shorter length, but no instances less than 20 seconds in length are included. Pain is predicted for short clips that are collated for video level pain prediction. We show results when using clips of length 1 frame (frame based), and with clips of length five seconds. The five second clip length is set following past research [229]; additionally, [223] suggests it to be the duration of time a horse pain expression lasts.

The backbone network is frozen when training the pain detection head, which is trained for 10 epochs at 0.001 learning rate. Leave one out training is again used, excluding the same test subject as for the backbone network. In addition pain detection performance on a validation set is calculated after each epoch, and the model at the epoch with the highest performance is used for testing. Data from the non-test subject with the most balanced pain/no-pain data distribution is used as the validation data.

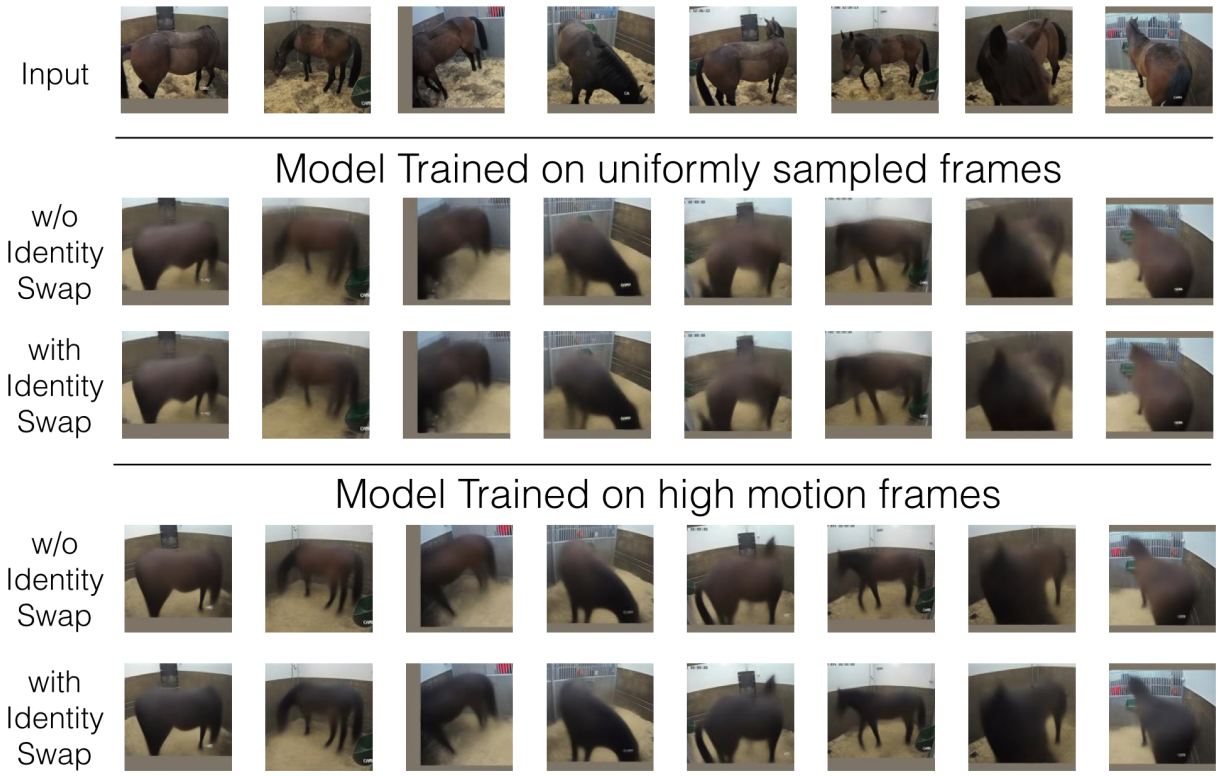

Figure 6.3: Identity swapping on different models. Each column shows the decoder output for the corresponding input image from the first row. In the third and fifth rows, the identity representations are swapped with the identity representation from a training horse with a black coat.

6.3.2 Disentangled Representation Learning

Disentangled Pose Representation

We explore the quality of the base network's latent representation qualitatively. The ideal pose representation would be able to cluster together the same horse pose regardless of viewpoint. In addition, the representation would be disentangled from horse identity.

Given the pose representation of a test input image at time t from i viewpoint, $p_{v^i,t}$, we find its top 3 nearest neighbors from the train data after rotation to viewpoint j, that is we find the nearest neighbors in the training data of $\mathbf{R}_{v^i \to v^j} p_{v^i,t}$. Some qualitative results are shown in Figure 6.4, where the second columns show the actual image from j viewpoint, $p_{v^j,t}$.

The top 3 neighbors are consistent with the expected ground truth, which shows that the latent representation has learned a pose representation that is viewpoint co-variant. One exception is the neighbors in the third row in the left set of columns – the second nearest neighbor is quite different from the ground truth image. On the other hand the nearest neighbors show evidence of both background and identity disentanglement. The background of the retrieved images are often different from the query background, for example in the middle set of columns. At the same time, the retrieved horses may be physically different from the query horse, for example a black horse is retrieved in the fourth row, in the left set of columns, and a horse with a white blaze is retrieved in the second row in the right set of columns. Interestingly, when the horse head and neck is self occluded in the second row, right set of columns, the nearest neighbors suggest that the model hallucinates a reasonable – though not entirely accurate – neck and head position.

Disentangled Identity Representation

In Figure 6.3, we show results of swapping the identity representation for a test horse. As explained in Section 6.2.2, the decoder uses an identity and pose representation to reconstruct an image. We compare reconstructed images with and without swapping the identity representations with the identity representation of a training horse with a black coat. Good disentanglement would show a horse with the same pose as the input image, but with a black rather than a brown coat.

The model trained with uniformly sampled video frames is not able to disentangle identity and pose and reconstructs horses with more or less the same color with and without identity swapping. However, by training on high motion frames, the identity is disentangled more fully, as can be seen by the horses in identity swapped images noticeably darker coats. This is because horses in the LPS dataset stay in the same pose for long periods of time, making it less likely that the swapped identity feature's input frame would feature a horse in a different pose from the pose feature's input frame when training with uniformly sampled frames than

	True Performance		Best Case Performance	
	F1 Score	Accuracy	F1 Score	Accuracy
Ours-Frame	**58.5±7.8**	**60.9±5.7**	60.8±6.4	62.3±5.4
Ours-Clip	55.9±5.1	57.8±4.4	**65.1±6.7**	**65.6±6.5**
Ours-Clip-HaS	56.5±5.0	58.6±4.3	63.6±6.2	64.6±5.8
Scratch	54.5±9.1	57.3±6.5	61.7±8.1	63.2±7.7

	F1 Score	Accuracy
Ours-MIL	**58.5±7.8**	**60.9±5.7**
CE-Clip	52.2±10.2	57.0±6.4
CE-Frame	49.1±10.9	55.2±5.9
MIL-OG	47.7±12.7	55.0±8.2

Table 6.1: Quantitative Results on Pain Detection. *Left*: Comparison of frame and clip based pain heads against a model trained from scratch with early stopping using a hold out dataset (Val Selected) and best case (Oracle). *Right*: Comparison of cross-entropy loss and multi instance learning loss variations.

with high motion frames.

Training with high motion frames also results in more fine grained reconstructions of a variety of poses. This can be seen by comparing the reconstructions, particularly around the head and legs, in the fifth and sixth columns. Lastly, the background is crisper in the last two rows. This is due to our use of background images that are derived from the same month as the input frame, and results in better quality reconstructions as the network learns to ignore background entirely.

6.3.3 Pain Detection

We present F1 score and accuracy for pain detection results. F1 score is the harmonic mean of precision and recall. We take the unweighted mean of the F1 score across both classes to evaluate each model. F1 scores are averaged across all training folds, and presented here alongside the standard deviation. We similarly present the raw accuracy scores averaged across all folds.

As stated in Section 6.2.3, we can train our model with both frame and clip (l consecutive frames) level inputs. The model predicts pain for each input, which is then collated for video level results. Clip level inputs allow the model to learn dynamic (temporal) features, but uses more parameters.

In Table 6.1(left) we show results of our frame and clip based model against a model that is trained from scratch. The scratch model has the same architecture as the encoder part of

the base network and the pain head and is trained on frames.

The 'True Performance' column shows the performance of the model selected by early stopping based on performance on a holdout validation set and shows the true performance. The 'Best Case Performance' column shows results on the test data if the stopping criteria aligned with the epoch with the best testing performance and shows the upper limit performance.

Both our frame and clip based models have better true performance than the model trained from scratch, with the frame based model showing 4% higher F1 score. Additionally, the best case performance is either comparable or better than the model trained from scratch, even though more parameters are learned specifically for the pain detection task in the 'Scratch' model. At the same time, our models' use of a disentangled pose representation ensures that only pose and not any extraneous information is used to deduce the pain state. These results indicate that using a disentangled pose representation is useful for a dataset such as ours with limited training subjects.

All models suffer from some degree of overfitting as can be seen from the difference between the true and best-case performance metrics. The model trained from scratch however, exhibits the highest amount of overfitting with more than 10% lower true F1 score than best case performance which can be expected since it learns the most number of parameters. While the true result is better with frame input than with video input, the use of temporal information through clip level inputs results in a much higher best-case performance, with 5% higher F1 best case evaluation. We therefore add more regularization by using random adversarial erasing on training clips as proposed in Hide-and-Seek (HaS) [230], which results in a 1% higher accuracy than without this augmentation method. The results indicate that clip based pain prediction is most promising, but would require more regularization to compete with the simple frame based results.

Weakly Supervised Learning

In these set of experiments, we evaluate the importance of our multi-instance learning (MIL) set up, with results in Table 6.1(right). As discussed in Section 6.2.3, our version of MIL loss (Ours-MIL), averages the pain and no-pain predictions of the top k time segments (or frames) with the highest *pain* prediction. We contrast this loss against the original MIL loss – MIL-OG – used in literature that averages the top pain and no-pain predictions separately to obtain the video level prediction. Lastly, we compare against a simple cross entropy loss – CE – where each frame or clip is separately supervised during training. The test results are still obtained by averaging the top k clip level predictions to keep results comparable.

Firstly, by comparing 'Ours-MIL' against 'CE' we see that using a weakly supervised setting is essential, and that pain and no-pain does not have a dense presence in this dataset. In fact, the results are close to random when a strongly supervised training model is used with F1 performance at 49.1% for 'CE-Frame'. The use of dynamic information with clip based inputs leads to improved performance, however, the overall performance is still lower than training with weak supervision.

Secondly, by comparing against 'MIL-OG', we see that our modified MIL loss is necessary to learn a reasonable model for pain. In fact, use of MIL-OG leads to a worse model of pain than random guessing with 47.7% F1 score. This bolsters our underlying reasoning that clips with no-pain features may exist in pain videos and should not be penalized during training in order to develop a good pain model.

6.3.4 Attributes of Pain

Figure 6.5 shows some clips that our model classifies as painful. The clips feature some classic signs of pain such as the 'lowered ears' [17] (second through fourth rows), a lifted hind leg (first row) which corresponds with 'non-weight bearing' [231], 'lying down' (fifth and sixth rows), 'looking at flank'(seventh row) as described in [232], and the gross pain behavior, 'stretching' (last row) [12]. These results show a good correspondence between

the visual attributes our model focuses on to determine pain, and the pain scales used by veterinary experts to determine equine pain.

6.4 Discussion

This work proposes a method for determining equine pain from surveillance footage, using weak labels. In order to ensure that pain is learned from horse body language, we use a self supervised generative model to disentangle horse pose from horse identity and background. The resulting pose representation is then used to learn a pain prediction model that is weakly supervised with a novel pain specific multi instance learning loss. Our models show a tendency to overfit, but can achieve performance up to 60% accuracy which is higher than human performance on equine pain detections as shown in a past study [28]. We qualitatively analyze our model's disentangled pose and identity features, and show quantitative and qualitative results on pain detection. We do not use any exclusion criteria for our test data, which often contains views of the horse that are cropped and self occluded, making successful visual determination of pain extremely unlikely. Future work should include a means to excluding pain predictions on video clips that do not have a clear enough view of the horse, and developing a means to regularize the pain detection head.

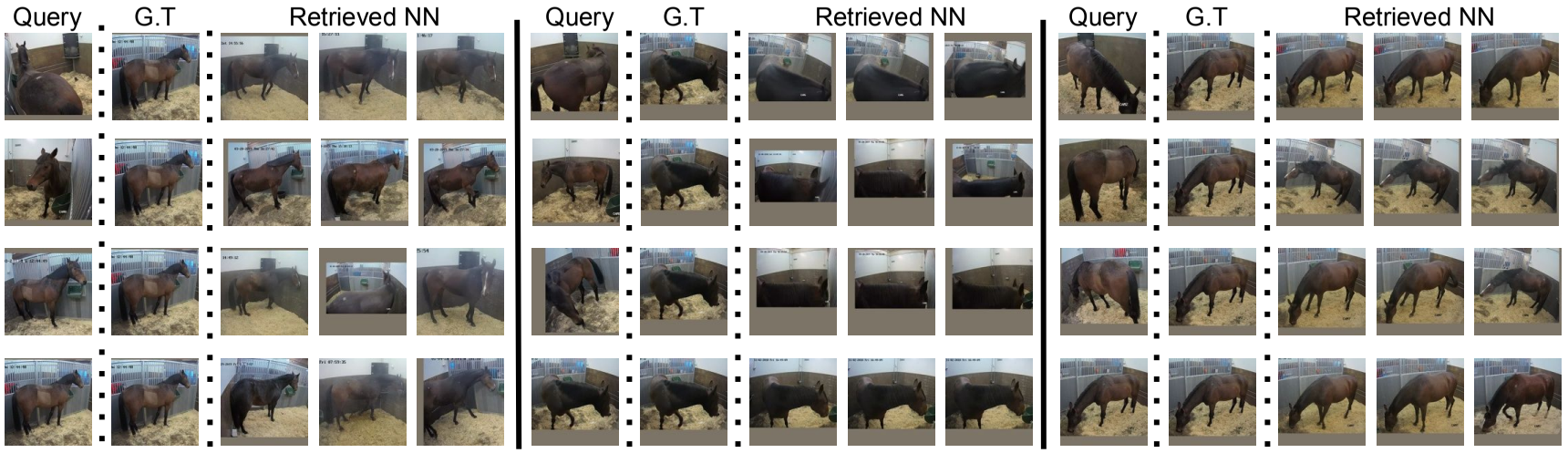

Figure 6.4: Nearest neighbor retrieval on latent pose representation. The pose representation of the query image is rotated before nearest neighbor retrieval. The nearest neighbors match the pose in the actual ground truth image from the rotated view.

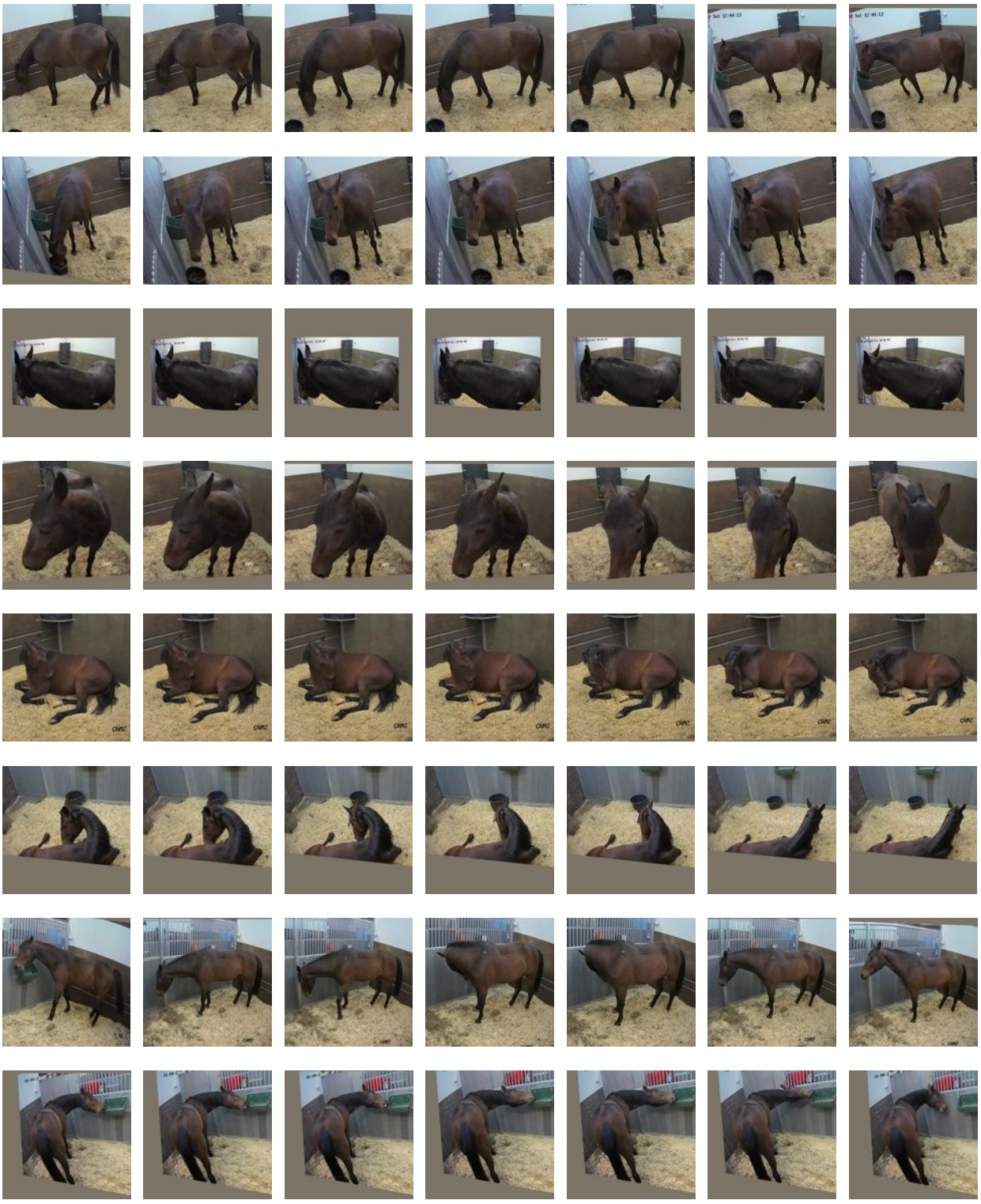

Figure 6.5: Video segments correctly detected as painful by our model. The detected segments display signs of pain such as avoiding weight bearing (1st row), backwards ears and painful facial expressions (2nd-4th rows), lying down (5-6th rows), looking at the painful leg (7th row), and stretching (last row).

Chapter 7

Conclusion

This thesis explored the problem of automatic visual detection of pain in the horse. As a cross-disciplinary and new area of research, it presented challenges that required interdisciplinary collaboration, and the invention of novel computer methods.

We first described the facial expression of pain in horses in terms of the objective, comprehensive, and biologically grounded language of equine facial action coding system. Towards this end, we developed a graph based method that uses correlations between facial movements to deduce the components of the pain expression in horses. Following, we developed an automatic and easy to use application for finding horse faces in videos that veterinary researchers may use to quickly identify time segments suitable for facial expression annotation from long videos. The application has saved EquiFACS annotators hours in valuable annotation time, and has been instrumental in the description of both the pain and the stress face in horses in terms of EquiFACS. These works have relied on interdisciplinary collaboration to push the boundaries of veterinary research in horse pain.

Apart from veterinary science, we also developed novel computer vision methods for identification of horse facial parts, weakly supervised action localization, and weakly supervised horse pain detection from surveillance video footage. We developed a means for identifying facial keypoints in animal faces that made use of large readily available human keypoint

datasets via face structure warping, and built and released a dataset for horse facial keypoint detection. We presented a graph convolution based method for action localization that, by the explicit use of similarity relationships between time segments in videos, could temporally localize the extent of actions in videos despite not being trained with any such annotation. Finally, we developed a method for pain detection in horses that used surveillance video footage with weak video level labels. In the process, we exploited the availability of multiple views of surveillance video to learn a representation of the horse pose that was independent of its background, and identity, and did not require any additional annotation. The pain detection model used horse pose cues exclusively to deduce the pain status of the horse, and identified pain features that aligned well with pain scales currently used by veterinary practitioners.

While this thesis has helped address some of the challenges in automatic horse pain detection, there are a number of directions in which research in this area can grow. Below, I summarize some of the exciting directions in which research in this field can and is growing.

7.1 Future Work

Beyond Pain. While pain is an extremely important modality of affective state in horses, it is also important to develop automated recognition systems for other emotional and behavioral states in horses. As prey animals, horses are prone to fear and anxiety, and being able to identify and prevent triggers for these negative affective states can not only improve horse well being, but reduce the chances of traumatic injury, and the substantial monetary cost of tranquilizers, relaxants, and even massage that horse practitioners currently incur [233]. In a recent work, we adapted the co-occurrence graph method from Chapter 2 to identify the expression of a negative, but not painful, affective state commonly referred to as stress in horses [46]. Future work would be creating an automated visual stress recognition system.

In addition, our understanding of pain expressions in horses is nascent. Similar to the

study by Prkatchin et al [35] on modalities of pain, future work can focus on understanding how horses express pain differently based on different sources and intensity of pain. Knowing this may be very useful for diagnosis of underlying medical conditions, particularly for those that are difficult to identify – such as low grade lameness. Relatedly, there is little work on the interaction, particularly temporal interaction, between facial and body expressions of pain, with previous work focusing on either faces (e.g. [27]) or body (Chapter 6). Understanding how both sources of expression correlate would be a useful, holistic, and consequently powerful direction of research, particularly since pain hiding under observation behavior probably relates the most to gross body behavior [19].

Beyond Videos and Images. This thesis has focused on image and video data. However, different modalities of information may be used to develop our understanding of horse pain. For instances heart monitors were used in the previously mentioned study of stress [46], and may also be used to study pain responses. Relatedly, 2D image and video data can be unflattened to get a more holistic and 3D understanding of the animal. This direction has been pursued to build 3D models of animals [215, 216], but can also be used to infer horse health, as has been done in [218] for deducing lameness. Audio and interactivity cues are also very important in veterinary pain evaluations, and may also be included in an automatic pain detection system for a more holistic understanding of pain.

Beyond Experimental Datasets. Expert collected experimental datasets like those used in Chapters 2 and 6 are important to develop a gold standard for both emotion understanding in animals, and emotion detection by automatic systems. However, these datasets are difficult and expensive to collect and cannot be scaled easily.

In this thesis, we have addressed the problem of dataset scalability in multiple ways. We presented methods that required only weak labels in Chapters 5 and 6, presented a method for assisting and speeding up annotation in Chapter 3, and proposed a means to transfer information from big datasets that are already available in Chapter 4.

Other works have also identified the need for scalable data solutions, and have like us, proposed to transfer information from human datasets [210], and from synthetic data [211].

There are millions of videos and images of horses on the web with tags about the horse type, age, and activity. Utilizing these resources can be extremely useful to build a better visual model of horse shape, and movement, and may even be useful for expression understanding. Crowd sourcing expression and face annotation may also be extremely useful to build a holistic understanding of animal facial expressions. In fact, crowd sourced annotation has been used for pig farms already in China [234] which makes the use of similar model for dataset creation an extremely promising direction for horse understanding.

Beyond Horses. Features of pain are shared amongst livestock mammals. For example orbital tightening has been described for horses [17], cattle [13], and sheep [14], and an arched back is a painful body behavior associated with both horses and cattle [232, 13]. It would be very useful to develop a cross-species pain detection model that is not only able to infer the pain status of different mammals, but is also able to train and transfer knowledge between the creatures. The method may draw from domain transfer as used in Chapter 4, as well as the use of self supervision for disentangled latent representation learning, as shown in Chapter 6, to project face and body behavior learned from each different species to a common latent space that is then jointly used to deduce pain state of the input livestock mammal.

Beyond Interdisciplinary Teams. An essential part of my PhD was learning to identify areas of research that were of interest to both me as a computer vision researcher, and my collaborators as veterinary researchers. It was important to understand how animal behavior studies are carried out, how video and image data from these studies is collected and stored, and finally, how it is annotated for analysis. It was also of great value to gain an overview of the statistical methods used to analyse collected data in veterinary science. Similarly, my veterinary collaborators learned how data is used to train, test, and evaluate machine

learning classification systems, and about the challenges computer vision systems face when working with small, or sparsely annotated datasets. While rewarding, these learning curves can be flattened in future interdisciplinary collaborations on automatic animal behavior understanding. This can include the development of courses presenting seminal research from across disciplines, course requirements that cut across research departments for graduate students, and co-advising by professors from both computer and veterinary or animal science. Relatedly, with the wide adoption of machine, particularly deep, learning in industry, easy to use machine learning tools are available for researchers without computer science background for use in their research work. Pairing student researchers with similar research interests, but different academic backgrounds can be a very productive means of growing research in automatic animal behavior detection.

Automatic recognition animal behavior is a challenging and multifaceted discipline that is only just beginning to emerge through collaborations between veterinary experts and computer vision and machine learning scientists. Even though this thesis presents first and small steps towards solving the larger problem of automatic behavior understanding in animals, I hope it will be helpful for future researchers in this domain.

Bibliography

[1] Marta E Alonso, José R González-Montaña, and Juan M Lomillos. Consumers' concerns and perceptions of farm animal welfare. *Animals*, 10(3):385, 2020. 2

[2] Rebecca Smithers. Third of britons have stopped or reduced eating meat - report. *The Guardian*. 2

[3] U.S. cow-calf production costs and returns per cow, 2008-2016. https://www.ers.usda.gov/data-products/commodity-costs-and-returns/. 2

[4] January 1 Cattle Inventory Up 2 Percent. http://usda.mannlib.cornell.edu/usda/current/Catt/Catt-01-31-2017.pdf. 2

[5] IASP Taxonomy: International Association for the Study of Pain; 2016. http://www.iasp-pain.org/Education/Content.aspx?ItemNumber=1698. Accessed: 2016-07-07. 2, 11

[6] V Molony and JE Kent. Assessment of acute pain in farm animals using behavioral and physiological measurements. *Journal of animal science*, 75(1):266–272, 1997. 2

[7] M Raekallio, PM Taylor, and M Bloomfield. A comparison of methods for evaluation of pain and distress after orthopaedic surgery in horses. *Veterinary Anaesthesia and Analgesia*, 24(2):17–20, 1997. 3, 11, 91

[8] Jill Price, Seago Catriona, Elizabeth M Welsh, and Natalie K Waran. Preliminary evaluation of a behaviour-based system for assessment of post-operative pain in horses

following arthroscopic surgery. *Veterinary anaesthesia and analgesia*, 30(3):124–137, 2003. 3, 11, 91

[9] Debra C Sellon, Malcolm C Roberts, Anthony T Blikslager, Catherine Ulibarri, and Mark G Papich. Effects of continuous rate intravenous infusion of butorphanol on physiologic and outcome variables in horses after celiotomy. *Journal of Veterinary Internal Medicine*, 18(4):555–563, 2004. 3, 11, 91

[10] C Graubner, V Gerber, M Doherr, and C Spadavecchia. Clinical application and reliability of a post abdominal surgery pain assessment scale (paspas) in horses. *The Veterinary Journal*, 188(2):178–183, 2011. 3, 11, 91

[11] Johannes PAM van Loon and Machteld C Van Dierendonck. Monitoring acute equine visceral pain with the Equine Utrecht University Scale for Composite Pain Assessment (EQUUS-COMPASS) and the Equine Utrecht University Scale for Facial Assessment of Pain (EQUUS-FAP): A scale-construction study. *The Veterinary Journal*, 206(3):356–364, 2015. 3, 5, 11, 28, 29, 37

[12] KB Gleerup and Casper Lindegaard. Recognition and quantification of pain in horses: A tutorial review. *Equine Veterinary Education*, 28(1):47–57, 2016. 3, 11, 91, 107

[13] Karina Bech Gleerup, Pia Haubro Andersen, Lene Munksgaard, and Björn Forkman. Pain evaluation in dairy cattle. *Applied Animal Behaviour Science*, 171:25–32, 2015. 3, 5, 29, 45, 114

[14] Krista M McLennan, Carlos JB Rebelo, Murray J Corke, Mark A Holmes, Matthew C Leach, and Fernando Constantino-Casas. Development of a facial expression scale using footrot and mastitis as models of pain in sheep. *Applied Animal Behaviour Science*, 176:19–26, 2016. 3, 114

[15] Abbie V Viscardi, Michelle Hunniford, Penny Lawlis, Matthew Leach, and Patricia V Turner. Development of a piglet grimace scale to evaluate piglet pain using facial

expressions following castration and tail docking: a pilot study. *Frontiers in veterinary science*, 4:51, 2017. 3

[16] Emanuela Dalla Costa, Michela Minero, Dirk Lebelt, Diana Stucke, Elisabetta Canali, and Matthew C Leach. Development of the Horse Grimace Scale (HGS) as a pain assessment tool in horses undergoing routine castration. *PLoS one*, 9(3):e92281, 2014. 3, 5, 11, 13, 27, 28, 29, 36, 49, 90

[17] Karina B Gleerup, Björn Forkman, Casper Lindegaard, and Pia H Andersen. An equine pain face. *Veterinary anesthesia and analgesia*, 42(1):103–114, 2015. 3, 11, 13, 14, 28, 35, 36, 45, 49, 90, 107, 114

[18] Polly M Taylor, Peter J Pascoe, and Khursheed R Mama. Diagnosing and treating pain in the horse: Where are we today? *Veterinary Clinics of North America: Equine Practice*, 18(1):1–19, 2002. 3

[19] Britt Alice Coles. *No Pain, More Gain? Evaluating Pain Alleviation Post Equine Orthopedic Surgery Using Subjective and Objective Measurement*. PhD thesis, Swedish University of Agricultural Sciences, Uppsala,Sweden, 2016. 3, 5, 90, 91, 113

[20] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 3, 56

[21] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018. 4

[22] Ke Zhang, Kristen Grauman, and Fei Sha. Retrospective encoders for video summarization. In *ECCV*, pages 383–399, 2018. 4

[23] Mei Wang and Weihong Deng. Deep face recognition: A survey. *arXiv preprint arXiv:1804.06655*, 2018. 4

[24] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 4

[25] Yiting Lu, Marwa Mahmoud, and Peter Robinson. Estimating sheep pain level using facial action unit detection. In *IEEE International Conference on Automatic Face & Gesture Recognition*, 2017. 4, 7, 38, 95

[26] Alexander H Tuttle, Mark J Molinaro, Jasmine F Jethwa, Susana G Sotocinal, Juan C Prieto, Martin A Styner, Jeffrey S Mogil, and Mark J Zylka. A deep neural network to assess spontaneous pain from mouse facial expressions. *Molecular pain*, 14:1744806918763658, 2018. 4, 95

[27] Hilde I Hummel, Francisca Pessanha, Albert Ali Salah, Thijs JPAM van Loon, and Remco C Veltkamp. Automatic pain detection on horse and donkey faces. In *FG*, 2020. 4, 7, 95, 113

[28] Sofia Broomé, Karina Bech Gleerup, Pia Haubro Andersen, and Hedvig Kjellström. Dynamics are important for the recognition of equine pain in video. *arXiv preprint arXiv:1901.02106*, 2019. 4, 6, 38, 90, 93, 95, 99, 108

[29] Gwen Littlewort, Jacob Whitehill, Tingfan Wu, Ian Fasel, Mark Frank, Javier Movellan, and Marian Bartlett. The computer expression recognition toolbox (cert). In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 298–305. IEEE, 2011. 4, 45

[30] Pooya Khorrami, Thomas Paine, and Thomas Huang. Do deep neural networks learn facial action units when doing expression recognition? In *CVPR Workshops*, 2015. 4

[31] Hui Ding, Shaohua Kevin Zhou, and Rama Chellappa. FaceNet2ExpNet: Regularizing a deep face recognition net for expression recognition. *arXiv preprint arXiv:1609.06591*, 2016. 4, 52

[32] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *2005 IEEE international conference on multimedia and Expo*, pages 5–pp. IEEE, 2005. 4

[33] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The Extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010. 4, 5

[34] Guoying Zhao, Xiaohua Huang, Matti Taini, Stan Z Li, and Matti PietikäInen. Facial expression recognition from near-infrared videos. *Image and Vision Computing*, 29(9):607–619, 2011. 4

[35] Kenneth M Prkachin. The consistency of facial expressions of pain: a comparison across modalities. *Pain*, 51(3):297–306, 1992. 4, 13, 31, 113

[36] Paul Ekman, Wallace V Friesen, and Joseph C Hager. Facial action coding system. manual and investigator's guide. 2002. 4, 12

[37] Josh M Susskind, Adam K Anderson, and Geoffrey E Hinton. The toronto face database. *Department of Computer Science, University of Toronto, Toronto, ON, Canada, Tech. Rep*, 3, 2010. 5

[38] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014. 5

[39] S Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013. 5

[40] C Fabian Benitez-Quiroz, Ramprakash Srinivasan, and Aleix M Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5562–5570, 2016. 5

[41] Kenneth D Craig and Christopher J Patrick. Facial expression during induced pain. *Journal of personality and social psychology*, 48(4):1080, 1985. 5

[42] Stuart A Grossman, Vivian R Sheidler, Karen Swedeen, John Mucenski, and Steven Piantadosi. Correlation of patient and caregiver ratings of cancer pain. *Journal of pain and symptom management*, 6(2):53–57, 1991. 6

[43] Jen Wathan, Anne M Burrows, Bridget M Waller, and Karen McComb. EquiFACS: The equine facial action coding system. *PloS one*, 10(8):e0131738, 2015. 6, 12, 15, 27, 28, 29, 34, 35, 37, 91

[44] Miriam Kunz, Doris Meixner, and Stefan Lautenbacher. Facial muscle movements encoding pain—a systematic review. *Pain*, 160(3):535–549, 2019. 6, 9, 13, 16

[45] Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *CVPR*, 2017. 6, 39

[46] Johan Lundblad, Maheen Rashid, Marie Rhodin, and Pia Haubro Andersen. Facial expressions of emotional stress in horses. *bioRxiv*, 2020. 6, 9, 112, 113

[47] Lauren R Finka, Stelio P Luna, Juliana T Brondani, Yorgos Tzimiropoulos, John McDonagh, Mark J Farnworth, Marcello Ruta, and Daniel S Mills. Geometric morphometrics for the study of facial expressions in non-human animals, using the domestic cat as an exemplar. *Scientific reports*, 9(1):1–12, 2019. 7

[48] Katrina Ask, Marie Rhodin, Lena-Mari Tamminen, Elin Hernlund, and Pia Haubro Andersen. Identification of body behaviors and facial expressions associated

with induced orthopedic pain in four equine pain scales. *Animals*, 10(11):2155, 2020. 9, 92, 96

[49] T Hadjistavropoulos and Kenneth D Craig. A theoretical framework for understanding self-report and observational measures of pain: a communications model. *Behaviour research and therapy*, 40(5):551–570, 2002. 11

[50] Paul Flecknell, Matthew Leach, and Melissa Bateson. Affective state and quality of life in mice. *Pain*, 152(5):963–964, 2011. 11

[51] Emanuela Dalla Costa, Riccardo Pascuzzo, Matthew C Leach, Francesca Dai, Dirk Lebelt, Simone Vantini, and Michela Minero. Can grimace scales estimate the pain status in horses and mice? a statistical approach to identify a classifier. *PloS one*, 13(8):e0200339, 2018. 12

[52] Michael A Sayette, Jeffrey F Cohn, Joan M Wertz, Michael A Perrott, and Dominic J Parrott. A psychometric evaluation of the facial action coding system for assessing spontaneous expression. *Journal of Nonverbal Behavior*, 25(3):167–185, 2001. 12

[53] Amy JD Hampton, Thomas Hadjistavropoulos, Michelle M Gagnon, Jaime Williams, and David Clark. The effects of emotion regulation strategies on the pain experience: a structured laboratory investigation. *Pain*, 156(5):868–879, 2015. 13

[54] Anna Julia Karmann, Christian Maihöfner, Stefan Lautenbacher, Wolfgang Sperling, Johannes Kornhuber, and Miriam Kunz. The role of prefrontal inhibition in regulating facial expressions of pain: a repetitive transcranial magnetic stimulation study. *The Journal of Pain*, 17(3):383–391, 2016. 13

[55] Eva G Krumhuber, Arvid Kappas, and Antony SR Manstead. Effects of dynamic aspects of facial expressions: A review. *Emotion Review*, 5(1):41–46, 2013. 13

[56] Casper Lindegaard, Maj H Thomsen, Stig Larsen, and Pia H Andersen. Analgesic efficacy of intra-articular morphine in experimentally induced radiocarpal synovitis in horses. *Veterinary anaesthesia and analgesia*, 37(2):171–185, 2010. 14

[57] Elan. *Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands*. 15

[58] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973. 19

[59] Katrina Merkies, Chloe Ready, Leanne Farkas, and Abigail Hodder. Eye blink rates and eyelid twitches as a non-invasive measure of stress in the domestic horse. *Animals*, 9(8):562, 2019. 28

[60] Sara Hintze, Samantha Smith, Antonia Patt, Iris Bachmann, and Hanno Würbel. Are eyes a mirror of the soul? what eye wrinkles reveal about a horse's emotional state. *PloS one*, 11(10):e0164017, 2016. 28

[61] Jennifer Wathan and Karen McComb. The eyes and ears are visual indicators of attention in domestic horses. *Current Biology*, 24(15):R677–R679, 2014. 29

[62] CC Caeiro, AM Burrows, and BM Waller. Development and application of catFACS: Are human cat adopters influenced by cat facial expressions? *Applied Animal Behaviour Science*, 189:66–78, 2017. 34

[63] Cátia C Caeiro, Bridget M Waller, Elke Zimmermann, Anne M Burrows, and Marina Davila-Ross. OrangFACS: A muscle-based facial movement coding system for orangutans (pongo spp.). *International Journal of Primatology*, 34(1):115–129, 2013. 34

[64] Dale J Langford, Andrea L Bailey, Mona Lisa Chanda, Sarah E Clarke, Tanya E Drummond, Stephanie Echols, Sarah Glick, Joelle Ingrao, Tammy Klassen-Ross, Michael L

LaCroix-Fralish, et al. Coding of facial expressions of pain in the laboratory mouse. *Nature methods*, 7(6):447–449, 2010. 34, 45, 49

[65] Rosenberg Ekman. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997. 35

[66] FH Ashley, AE Waterman-Pearson, and HR Whay. Behavioural assessment of pain in horses and donkeys: application to clinical practice and future studies. *Equine veterinary journal*, 37(6):565–575, 2005. 35

[67] B. Coles, L. Birgitsdottir, and P. H. Andersen. Out of sight but not out of clinician's mind: Using remote video surveillance to disclose concealed pain behavior in hospitalized horses. In *International Association for the Study of Pain 17th World Congress*, 2018. 35

[68] Sue Dyson, Jeannine M Berger, Andrea D Ellis, and Jessica Mullard. Can the presence of musculoskeletal pain be determined from the facial expressions of ridden horses (FEReq)? *Journal of veterinary behavior*, 19:78–89, 2017. 36

[69] Vidit Jain and Erik Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, 2010. 37

[70] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. *arXiv preprint arXiv:1511.06523*, 2015. 37

[71] Martin Koestinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *BeFIT Workshop*, 2011. 37, 50, 57, 59

[72] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks). In *ICCV*, 2017. 37

[73] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001. 37

[74] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. DSFD: Dual shot face detector. *arXiv preprint arXiv:1810.10220*, 2018. 37

[75] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *CVPR*, 2018. 37

[76] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 37

[77] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 37, 39, 58

[78] Heng Yang, Renqiao Zhang, and Peter Robinson. Human and sheep facial landmarks localisation by triplet interpolated features. In *WACV*, 2016. 37, 50, 53, 59, 60, 61, 62, 63

[79] Maheen Rashid, Xiuye Gu, and Yong Jae Lee. Interspecies knowledge transfer for facial keypoint detection. In *CVPR*, 2017. 37, 39, 41, 95

[80] Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017. 37

[81] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In *CVPR*, 2018. 37

[82] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards high performance video object detection. In *CVPR*, 2018. 44

[83] M Arkell, RM Archer, FJ Guitian, and SA May. Evidence of bias affecting the interpretation of the results of local anaesthetic nerve blocks when assessing lameness in horses. *Veterinary Record*, 159(11):346–348, 2006. 45

[84] Kris Descovich, Jennifer Wathan, Matthew C Leach, Hannah M Buchanan-Smith, Paul Flecknell, David Farningham, and Sarah-Jane Vick. Facial expression: An under-utilised tool for the assessment of welfare in mammals. *Altweb (Johns Hopkins Center for Alternatives to Animal Testing (CAAT))*, 2017. 45

[85] Ruth VE Grunau and Kenneth D Craig. Pain expression in neonates: facial action and cry. *Pain*, 28(3):395–410, 1987. 45

[86] Miriam Kunz and Stefan Lautenbacher. The faces of pain: a cluster analysis of individual differences in facial activity patterns of pain. *European Journal of Pain*, 18(6):813–823, 2014. 45

[87] Marian Stewart Bartlett, Gwen C Littlewort, Mark G Frank, and Kang Lee. Automatic decoding of facial movements reveals deceptive pain expressions. *Current Biology*, 24(7):738–743, 2014. 45

[88] A Boissy, A Aubert, L Désiré, L Greiveldinger, E Delval, and I Veissier. Cognitive sciences to relate ear postures to emotions in sheep. *Animal Welfare*, 20(1):47, 2011. 49

[89] E Holden, G Calvo, M Collins, A Bell, J Reid, EM Scott, and AM Nolan. Evaluation of facial expression in acute pain in cats. *Journal of Small Animal Practice*, 55(12):615–621, 2014. 49

[90] Xi Peng, Rogerio S Feris, Xiaoyu Wang, and Dimitris N Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *ECCV*, 2016. 49, 52

[91] Shengtao Xiao, Jiashi Feng, Junliang Xing, Hanjiang Lai, Shuicheng Yan, and Ashraf Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *ECCV*, 2016. 49, 52

[92] George Trigeorgis, Patrick Snape, Mihalis A Nicolaou, Epameinondas Antonakos, and Stefanos Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *CVPR*, 2016. 49, 52

[93] Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Occlusion-free face alignment: Deep regression networks coupled with de-corrupt autoencoders. In *CVPR*, 2016. 49, 52

[94] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3D model fitting. In *CVPR*, 2016. 49, 52

[95] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3D solution. In *CVPR*, 2016. 49, 52

[96] Dong Chen, Gang Hua, Fang Wen, and Jian Sun. Supervised transformer network for efficient face detection. In *ECCV*, 2016. 49, 52, 56

[97] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *arXiv preprint arXiv:1604.02878*, 2016. 49, 52

[98] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 50

[99] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 50

[100] Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, et al. Active appearance models. *TPAMI*, 23(6):681–685, 2001. 52

[101] Iain Matthews and Simon Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 52

[102] Jason Saragih and Roland Goecke. A nonlinear discriminative approach to AAM fitting. In *ICCV*, 2007. 52

[103] Georgios Tzimiropoulos and Maja Pantic. Optimization problems for fast AAM fitting in-the-wild. In *CVPR*, 2013. 52

[104] David Cristinacce and Timothy F Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006. 52

[105] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008. 52

[106] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011. 52

[107] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013. 52

[108] Michel Valstar, Brais Martinez, Xavier Binefa, and Maja Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, 2010. 52

[109] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 52

[110] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *IJCV*, 107(2):177–190, 2014. 52

[111] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *CVPR*, 2015. 52

[112] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *CVPR*, 2010. 52

[113] Donghoon Lee, Hyunsin Park, and Chang D Yoo. Face alignment using cascade gaussian process regression trees. In *CVPR*, 2015. 52

[114] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, 2015. 52

[115] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *PAMI*, 35(12):2930–2940, 2013. 52

[116] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Unconstrained face alignment via cascaded compositional learning. In *CVPR*, 2016. 52

[117] Oncel Tuzel, Salil Tambe, and Tim K Marks. Robust face alignment using a mixture of invariant experts. In *ECCV*, 2016. 52

[118] Feng Liu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. Joint face alignment and 3d face reconstruction. In *ECCV*, 2016. 52

[119] Dong Chen, Shaoqing Ren, Yichen Wei, Xudong Cao, and Jian Sun. Joint cascade face detection and alignment. In *ECCV*, 2014. 52

[120] Yue Wu and Qiang Ji. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In *CVPR*, 2016. 52

[121] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013. 52, 57, 59

[122] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCV Workshops*, 2013. 52

[123] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, 2014. 52

[124] Zhujin Liang, Shengyong Ding, and Liang Lin. Unconstrained facial landmark localization with backbone-branches fully-convolutional networks. *arXiv preprint arXiv:1507.03409*, 2015. 52

[125] Yue Wu and Tal Hassner. Facial landmark detection with tweaked convolutional neural networks. *arXiv preprint arXiv:1511.04031*, 2015. 52, 57

[126] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 52

[127] Heng Yang, Wenxuan Mou, Yichi Zhang, Ioannis Patras, Hatice Gunes, and Peter Robinson. Face alignment assisted by head pose estimation. *arXiv preprint arXiv:1507.03148*, 2015. 52

[128] Xiang Yu, Feng Zhou, and Manmohan Chandraker. Deep deformation network for object landmark localization. In *ECCV*, 2016. 52

[129] Iacopo Masi, Anh Tuan Tran, Jatuporn Toy Leksut, Tal Hassner, and Gerard Medioni. Do we really need to collect millions of faces for effective face recognition? *arXiv preprint arXiv:1603.07057*, 2016. 52

[130] Saurabh Singh, Derek Hoiem, and David Forsyth. Learning to localize little landmarks. In *CVPR*, 2016. 53

[131] Kevin J Shih, Arun Mallya, Saurabh Singh, and Derek Hoiem. Part localization using multi-proposal consensus for fine-grained categorization. *BMVC*, 2015. 53

[132] Jiongxin Liu, Yinxiao Li, and Peter N Belhumeur. Part-pair representation for part localization. In *ECCV*, 2014. 53

[133] Jiongxin Liu and Peter N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *ICCV*, 2013. 53

[134] F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *TPAMI*, 11(6):567–585, 1989. 56

[135] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 56, 59

[136] Krishna Kumar Singh and Yong Jae Lee. End-to-end localization and ranking for relative attributes. In *ECCV*, 2016. 56

[137] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. *CVPR*, 2016. 56

[138] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 57

[139] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 58

[140] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 61

[141] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011. 68

[142] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017. 68

[143] Yuan Yuan, Yueming Lyu, Xi Shen, Ivor W Tsang, and Dit-Yan Yeung. Marginalized average attentional network for weakly-supervised learning. In *ICLR*, 2019. 69, 70, 71, 81

[144] Sujoy Paul, Sourya Roy, and Amit K. Roy-Chowdhury. W-TALC: Weakly-supervised temporal activity localization and classification. In *ECCV*, 2018. 69, 70, 71, 74, 75, 76, 77, 81, 83

[145] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 2018. 69, 70, 71, 81

[146] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017. 69, 70, 71, 75, 76, 81, 101

[147] Zhi-Hua Zhou. Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep*, 2004. 70

[148] Yunlu Xu, Chengwei Zhang, Zhanzhan Cheng, Jianwen Xie, Yi Niu, Shiliang Pu, and Fei Wu. Segregated temporal assembly recurrent networks for weakly supervised multiple action detection. In *AAAI*, 2019. 70, 72, 81, 82

[149] N Kipf Thomas and Max Welling. Semi-supervised classification with graph convolutional networks. arxiv preprint. *arXiv preprint arXiv:1609.02907*, 103, 2016. 70, 72

[150] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 71, 79

[151] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 71, 79

[152] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 71, 80, 81

[153] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 71, 81

[154] Runhao Zeng, Chuang Gan, Peihao Chen, Wenbing Huang, Qingyao Wu, and Mingkui Tan. Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization. *IEEE Transactions on Image Processing*, 2019. 71

[155] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. AutoLoc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 2018. 71, 81

[156] Yuanhao Zhai, Le Wang, Ziyi Liu, Qilin Zhang, Gang Hua, and Nanning Zheng. Action coherence network for weakly supervised temporal action localization. In *ICIP*, 2019. 71

[157] Parthipan Siva and Tao Xiang. Weakly supervised action detection. In *BMVC*, 2011. 72

[158] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Towards weakly supervised action localization. *arXiv preprint arXiv:1605.05197*, 2016. 72

[159] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 72

[160] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 72

[161] Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *ICCV*, 2013. 72

[162] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *CVPR*, 2018. 72

[163] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 72

[164] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 72

[165] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016. 72

[166] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *Computer Vision and Image Understanding*, 163:78–89, 2017. 72

[167] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017. 72

[168] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what? - anticipating temporal occurrences of activities. In *CVPR*, 2018. 72

[169] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *CVPR*, 2018. 72

[170] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 72

[171] Yuan Yuan, Xiaodan Liang, Xiaolong Wang, Dit-Yan Yeung, and Abhinav Gupta. Temporal dynamic graph LSTM for action-driven video object detection. In *ICCV*, 2017. 72

[172] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 72

[173] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 72

[174] Pallabi Ghosh, Yi Yao, Larry S. Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018. 72

[175] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. *arXiv preprint arXiv:1811.12814*, 2018. 72

[176] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. *arXiv preprint arXiv:1812.03544*, 2018. 72

[177] Hao Huang, Luowei Zhou, Wei Zhang, and Chenliang Xu. Dynamic graph modules for modeling higher-order interactions in activity recognition. *arXiv preprint arXiv:1812.05637*, 2018. 72

[178] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 74, 80

[179] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 80

[180] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *ICCV*, 2017. 81

[181] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018. 81

[182] Aj Piergiovanni and Michael Ryoo. Temporal gaussian mixture layer for videos. In *ICML*, 2019. 81

[183] Pia H Andersen, KB Gleerup, J Wathan, B Coles, H Kjellström, S Broomé4 YJ Lee, M Rashid, C Sonder, E Rosenberg, and D Forster. Can a machine learn to see horse pain? an interdisciplinary approach towards automated decoding of facial expressions of pain in the horse. 90

[184] Patrick Lucey, Jeffrey F Cohn, Kenneth M Prkachin, Patricia E Solomon, and Iain Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. In *Face and Gesture 2011*, pages 57–64. IEEE, 2011. 91

[185] 91

[186] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 92, 94

[187] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 92, 94

[188] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 750–767, 2018. 92, 93, 94, 97, 98, 99, 102

[189] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7781–7790, 2019. 94

[190] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 94

[191] David Novotny, Ben Graham, and Jeremy Reizenstein. Perspectivenet: A scene-consistent image generator for new view synthesis in real indoor environments. In *Advances in Neural Information Processing Systems*, pages 7601–7612, 2019. 94

[192] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 94

[193] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 94

[194] Shubham Tulsiani, Saurabh Gupta, David F Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 302–310, 2018. 94

[195] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2172–2182, 2019. 94

[196] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 94

[197] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 94

[198] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 94

[199] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 94

[200] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6490–6499, 2019. 94

[201] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000. 94

[202] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29:2172–2180, 2016. 94

[203] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*. 94

[204] Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017. 94

[205] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3399–3407, 2018. 94

[206] Helge Rhodin, Victor Constantin, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. Neural scene decomposition for multi-person motion capture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7703–7713, 2019. 94

[207] Xipeng Chen, Kwan-Yee Lin, Wentao Liu, Chen Qian, and Liang Lin. Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 94

[208] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-

supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 94

[209] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 94

[210] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9498–9507, 2019. 95, 114

[211] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12386–12395, 2020. 95, 114

[212] Heng Yang, Renqiao Zhang, and Peter Robinson. Human and sheep facial landmarks localisation by triplet interpolated features. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016. 95

[213] Muhammad Haris Khan, John McDonagh, Salman Khan, Muhammad Shahabuddin, Aditya Arora, Fahad Shahbaz Khan, Ling Shao, and Georgios Tzimiropoulos. Animalweb: A large-scale hierarchical dataset of annotated animal faces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 95

[214] Artsiom Sanakoyeu, Vasil Khalidov, Maureen S. McCarthy, Andrea Vedaldi, and Natalia Neverova. Transferring dense pose to proximal animal classes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 95

[215] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 95, 113

[216] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3955–3963, 2018. 95, 113

[217] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images "in the wild". In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5358–5367. IEEE. 95

[218] Ci Li. Automatic horse lameness detection through 2d to 3d reconstruction, 2020. 95, 113

[219] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 97

[220] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 97

[221] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 98

[222] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 99

[223] Maheen Rashid, Alina Silventoinen, Karina Bech Gleerup, and Pia Haubro Andersen. Equine facial action coding system for determination of pain-related facial responses in videos of horses. *bioRxiv*, 2020. 99, 102

[224] Sujoy Paul, Sourya Roy, and Amit K. Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 101

[225] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018. 101

[226] Maheen Rashid, Hedvig Kjellstrom, and Yong Jae Lee. Action graphs: Weakly-supervised action localization with graph convolution networks. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 615–624, 2020. 101

[227] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask rcnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 102

[228] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003. 102

[229] Sofia Broomé, Karina Bech Gleerup, Pia Haubro Andersen, and Hedvig Kjellström. Dynamics are important for the recognition of equine pain in video. *arXiv preprint arXiv:1901.02106*, 2019. 102

[230] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3524–3533, 2017. 106

[231] G Bussieres, C Jacques, O Lainay, G Beauchamp, Agnès Leblond, J-L Cadoré, L-M Desmaizières, SG Cuvelliez, and E Troncy. Development of a composite orthopaedic pain scale in horses. *Research in veterinary science*, 85(2):294–306, 2008. 107

[232] Lori C Pritchett, Catherine Ulibarri, Malcolm C Roberts, Robert K Schneider, and Debra C Sellon. Identification of potential physiological and behavioral indicators of postoperative pain in horses after exploratory celiotomy for colic. *Applied Animal Behaviour Science*, 80(1):31–43, 2003. 107, 114

[233] Jill Stowe. 2018 american horse publications (ahp) equine industry survey sponsored by zoetis. *American Horse Publications*, 2018. 112

[234] Xiaowei Wang. Behind china's 'pork miracle': how technology is transforming rural hog farming. *The Guardian*. 114