Dynamics are Important for the Recognition of Equine Pain in Video

Sofia Broomé[1] Karina Bech Gleerup[2] Pia Haubro Andersen[3] Hedvig Kjellström[1]
[1] Division of Robotics, Perception, and Learning, KTH Royal Institute of Technology, Sweden
[2] Section for Medicine and Surgery, University of Copenhagen, Denmark
[3] Deptartment of Clinical Sciences, Swedish University of Agricultural Sciences, Sweden
sbroome@kth.se kbg@sund.ku.dk pia.haubro.andersen@slu.se hedvig@kth.se

Abstract

A prerequisite to successfully alleviate pain in animals is to recognize it, which is a great challenge in non-verbal species. Furthermore, prey animals such as horses tend to hide their pain. In this study, we propose a deep recurrent two-stream architecture for the task of distinguishing pain from non-pain in videos of horses. Different models are evaluated on a unique dataset showing horses under controlled trials with moderate pain induction, which has been presented in earlier work. Sequential models are experimentally compared to single-frame models, showing the importance of the temporal dimension of the data, and are benchmarked against a veterinary expert classification of the data. We additionally perform baseline comparisons with generalized versions of state-of-the-art human pain recognition methods. While equine pain detection in machine learning is a novel field, our results surpass veterinary expert performance and outperform pain detection results reported for other larger non-human species.

1. Introduction

This paper presents a method for automatic detection of equine pain behavior in video – a highly challenging problem since horses display pain through subtle signals.

Recognition of pain in animals is important because pain compromises animal welfare and can be a manifestation of disease. Pain diagnostics for humans typically include self-evaluation with the help of standardized forms and labeling of the pain by a clinical expert using pain scales. However, animals cannot verbalize their pain. The use of standardized pain scales is moreover challenged by the fact that prey animals such as horses and cattle display subtle and less obvious pain behavior. It is simply beneficial for a prey animal to appear healthy, to lower the interest from predators.

Pain in horses is instead typically assessed by (manually) observing a combination of behavioral traits such as exploratory behavior, restlessness, positioning in the box and changes in facial expressions [37]. Video filming may be used to prolong the observation periods and increase the likelihood of capturing pain behavior. However, proper manual annotation such as EquiFACS [37] of the facial expressions and other behavioral signs of pain is time consuming [19] – it can take two hours to evaluate a two minute video clip. This severely limits the possibilities for routine pain diagnostics on horses, for instance before or after surgery. An automatic pain detection system would allow veterinarians and horse owners to reliably and frequently screen the pain level of the horse. Section 2.1 provides more background on equine pain from a veterinary point of view.

We introduce a new research direction and application field in computer vision: detection of equine pain in video. While human pain detection has been performed with single images, owing to our expressive faces, the research hypothesis in this paper is that the pose and movement patterns matter more for horses. The equine pain detection task is explored using recurrent neural network architectures. These memory-preserving networks are chosen in order to model temporality, which is considered crucial when assessing pain in horses [37]. We have not seen previous work in pain detection for non-human species where temporal information is taken into account.

Specifically, we propose a deep neural network approach to automatic detection of horse pain in video, taking both spatial and temporal context into account (Section 4). Our architecture is a new combination of existing network types, using optical flow as an attention mask in a Convolutional LSTM [29] two-stream setting. We have reinterpreted the two-stream network first presented by [30] into a fully recurrent two-stream Convolutional LSTM network. The fusion of the two streams is done on the feature-level, adding the optical flow feature map to the RGB feature map element-wise, which highlights the regions of interest in the video. The method learns patterns end-to-end without the help of intermediate representations such as the Facial Action Coding System (FACS) [5]. To the best of our knowledge, this has not been done previously for animals.

The method is experimentally shown to outperform several baselines, including a method relying on single-frame analysis. Most notably, we show that our method performs better than manual classification by veterinarians with extensive expertize in equine pain assessment.

2. Background and related work

2.1. Pain evaluation in horses

Within the last decade, there has been an increasing focus on pain evaluation in horses, both because there is an increasing knowledge about the detrimental effects of pain and because advanced treatments and surgical procedures have become more common for horses [9]. Compared to small animals, horses are often not given the optimal pain treatment [15]. Reducing pain to a minimum ensures animal welfare, improves convalescence and optimizes treatment success [26]. Pain detection and quantification in horses depend on an observer to detect potentially pain related changes in behavior and in physiological parameters.

There is increasing focus on behavioral observations, since physiological parameters are often not pain-specific [1]. When in pain, horses change their activity budget [23, 24], meaning the percentage of their time that they spend lying down, eating, being attentive, etc. However, documenting this is very time consuming. Even though short, applicable pain scoring tools have been developed [9], these do not capture all pain behavior but rather do spot sampling. An automated means to record pain behavior from surveillance videos would be an important step forward in equine pain research.

Facial expressions have lately been presented as a sensitive measure of pain in horses, when observed in combination with other pain behaviors [3, 8, 9, 35]. A few examples of the facial pain characteristics described in [8] are slightly dropped ears, muscle tension around the eyes with an increased incidence of exposing the white in the eye, a square-like shape of the nostrils and an increased tension of the lips and chin.

2.2. Automated pain recognition

Automated pain recognition has largely focused on human pain. Littlewort et al. [14] distinguish between posed and real pain for humans in video by training separate linear support vector machine (SVM) classifiers for different FACS [5] action units. Among the works on pain recognition in humans using deep learning are Kahou et al. [11], Rodriguez et al. [25] and Zhou et al. [38]. [25] use an LSTM layer on top of a VGG-16 [31] CNN. Their task is to detect pain in the UNBC-McMaster Shoulder Pain Database [17], which is a video dataset for human pain recognition. Features up until and including the first dense layer in VGG-16 are extracted frame-wise and then arranged in sequences that are fed to the LSTM. The top LSTM layer is found to improve the result significantly compared to the base model. By aligning, cropping, masking and frontalizing the input data to the VGG+LSTM architecture they obtain the best known result on this dataset. Albeit without the preprocessing, we have included tests of their architecture on our dataset in Section 5.

Lu et al. [16] explore automated pain assessment in sheep faces from still images. To our knowledge, this is the only previous work in automatic recognition of pain for larger animals. Facial action units associated with pain are detected through the successive steps of facial detection, landmark detection, feature extraction and classification with an SVM. Our system performs better than the one of [16] on the binary pain detection task for unseen subjects.

Tuttle et al. [34] use deep learning to perform automatic facial action unit detection in mice, to score for pain. The single-frame InceptionV3 CNN is trained to detect action units, which are counted according to a grimace scale to establish the presence of pain.

Differently from the work of [16] and [34], our method takes an end-to-end approach to the problem, so as to not depend on pre-defined pain cues. We do not only study the face but also the pose, body and movement pattern of the animal, which we believe is crucial for reliable pain diagnostics in horses. The data both in [16] and in [34] are labeled according to a grimace scale ground truth and not according to whether pain was induced or not. Furthermore, the respective datasets are both culled so that only clear frontal faces are considered.

2.3. Action recognition in video

Action recognition methods can guide model development for pain recognition in video data, since both fields require the capability of extracting information from sequences of images. The challenge in action recognition and computer vision for video lies in fully using the information captured in dependencies across the temporal dimension of the data. The task of recognizing movements and gestures from video is still far from human performance, in contrast to single image object recognition. There are different ways of utilizing temporal information across a sequence of images. In a neural network context, this can for example be achieved using 3D convolutional layers, pooling across the temporal dimension, or by using recurrence. Long Short-Term Memory (LSTM) networks [10] are Recurrent Neural Networks (RNNs), proven to work well for data with long-term dependencies since they can mitigate the problem of vanishing gradients, from which standard RNNs suffer [22].

An early work in action recognition using deep learning by Karpathy et al. [12] compares single-frame and sequential models and finds only a marginal difference between the two approaches. But more recently, sequential models have clearly outperformed the ones that are trained on single frames [2, 21, 36]. Single-frame models base their classification on image content instead of on the dynamics between frames. This works to some extent, especially on datasets such as Sports-1M [12] or UCF-101 [32] where most of the activity categories contain pertinent visual objects – consider for example walking a dog or playing the piano.

Ng et al. [21] mainly test two approaches to the sequential activity recognition problem, after deep two-stream CNN [30] feature extraction of the frames and their corresponding optical flow. Stacked LSTM layers are compared to different ways of feature pooling across the temporal dimension. The former approach obtained the thereto highest result on the Sports-1M dataset.

Concerning the two-stream model, in [11] a comparison is made between stream fusion at the feature-level vs. fusion at the decision-level (meaning typically after the softmax output), and the authors find that feature-level fusion improves their results. In our two-stream architecture experiments, we follow their example and fuse the streams at the feature-level.

Shi et al. [29] introduce the Convolutional LSTM (C-LSTM) unit, where the fully-connected matrix multiplications involving the weight matrices in the LSTM equations [10] are replaced with convolutions. The idea with the C-LSTM is to preserve spatial structure and avoid having to collapse the input images to vectors, as one needs to do for a standard LSTM (Section 4.1). Networks composed of C-LSTM units were the best-performing architectures in our experiments on the equine dataset.

The C-LSTM in [29] is applied to weather prediction with radar map sequences as input. But owing to the contribution of [29], C-LSTM layers have recently been used in a few works on action and gesture recognition from video data, notably by Li et al. [13] and Zhu et al. [39]. The architecture of [39] consists of two streams: one using RGB and one using RGB-D as input. The streams are each composed of a 4-layer 3D CNN followed by a 2-layer C-LSTM network, spatial pyramid pooling and a dense layer before fusion. Similar to our method, [39] perform simultaneous extraction of temporal and spatial features. Their idea is to let the 3D convolutions handle the short-term temporal dependencies of the data and the C-LSTM the longer-term dependencies. We differ from the method of [39] in that we use optical flow and not RGB-D data for the motion stream and that our parallel streams consist only of recurrent C-LSTM layers, without 3D convolutions.

Table 1: Dataset overview, per horse subject and pain label (hh:mm:ss).

Subj. ID	1	2	3	4	5	6	Total
Gender	Mare	Mare	Mare	Mare	Mare	Gelding	n/a
Age (years)	14	7	12	6	14	3	n/a
Pain	00:58:55	00:39:30	00:29:52	00:22:21	00:44:38	00:26:09	03:41:25
No pain	01:00:27	01:01:29	01:01:10	01:31:12	00:59:49	00:29:37	06:03:44
Total	01:59:22	01:40:59	01:31:02	01:53:33	01:44:27	00:55:46	09:45:09
# frames, 2 fps	14324	12118	10924	13626	12534	6692	70292

Li et al. [13] introduce the VideoLSTM architecture, using C-LSTM networks in combination with a soft attention mechanism both for activity recognition and localization. Features are first extracted from RGB and optical flow frames using a single-frame VGG-16 [31] pre-trained on ImageNet [4] before they are fed to the recurrent C-LSTM streams. The attention in [13] is computed by convolving the input feature map at t with the hidden state from $t - 1$. A given C-LSTM stream in their pipeline always has one layer, consisting of 512 units. By contrast, our C-LSTM network is deep but has fewer units per layer, and we do not separate feature extraction from the temporal modeling.

3. The Equine Pain Dataset

The equine pain dataset used in this paper has been collected by Gleerup et al. [8], and consists of 9 hours and 45 minutes of video across six horse subjects, out of which 3 hours and 41 minutes are labeled as pain and 6 hours and 3 minutes as non-pain. These binary labels have been set according to the presence of pain induction, known from the recording protocol (Section 3.1). Frames are extracted from the equine videos at 2 fps. When training on 10-frame sequences, the dataset contains $\sim$ 7k sequences. Duration-wise this dataset is comparable to known video datasets such as Hollywood2 [18] (20 hours) and UCF-101 [32] (30 hours). Details on the class distribution across the different subjects can be seen in Table 1. A more complete description with image examples can be found in [8].

3.1. Recording setup and pain induction

The subjects were trained with positive reinforcement before the recordings to be able to stand relatively still in the trial area. They were filmed during positive and negative pain induction trials with a stationary camera at a distance of approximately two meters. The imagery contains the head and the body until the wither. The camera was fixed, which means that the data is suited for extraction of optical flow, since there is no camera motion bias [12].

Pain was induced using one out of two noxious stimuli applied to the horses for 20 minutes: a pneumatic blood pressure cuff placed around the antebrachium or the application of capsaicin 10% (chili extract) on 10 cm^2 skin [8]. Both types of experimental pain are ethically regulated and also occur in human pain research. They caused moderate but completely reversible pain to the horses [8].

3.2. Noise and variability

The videos in the dataset present some challenging noise. The lighting conditions were not ideal and some videos are quite dark. Unexpected elements like a veterinarian standing by the horse for a while, frequent occlusion of the horse's head, varying backgrounds, poses, camera angles and colors on halters all contribute to the challenging nature of the dataset. The coat color of the horses is mostly dark brown except for one chestnut but they do have varying characteristics such as stars and blazes.

Furthermore, there is some variability among the subjects. Subject 6 is by far the youngest horse at age 3, and frequently moves in and out of the frame. This presented difficulties for our classifiers as can be seen in Table 4. Importantly, this subject did not go through the same training for standing still as the other subjects. We include test runs on this horse for transparency.

4. Approach

Three main architectures are investigated, and each is additionally extended to a two-stream version. With two streams, we can feed sequences of different modalities to the network. In our case, we use RGB in one stream and optical flow in the other. The optical flow of two adjacent frames is an approximation of the gradients (u and v) of the pixel trajectories in the horizontal and vertical directions, respectively. We compute the optical flow using the algorithm presented by [6] and add the magnitude of u and v as a third channel to the tensor. The one-stream models receive either single frames or sequences of either modality.

The three base architectures are the deep CNN InceptionV3 [33], which takes single-frame input, the partly recurrent VGG+LSTM architecture from [25] which is a VGG-16 CNN [31] up to and including the first dense layer with one LSTM layer on top, and the fully recurrent C-LSTM. Intuitively, the three models differ in the extent by which they can model the dynamics of the data. InceptionV3 is a static model that only learns patterns from single frames. In its two-stream setting, it processes single optical flow frames in parallel to the RGB frames, which adds limited motion information. The VGG+LSTM model extracts temporal features, but does so separately from the spatial features. Its recurrent top layer is not an integral part of the network, as it is for the C-LSTM model. In the C-LSTM architecture all layers are fully recurrent and the spatial and temporal feature extraction take place simultaneously.

4.1. Convolutional LSTM

Shi et al. [29] introduce the Convolutional LSTM (C-LSTM) unit, where the fully-connected matrix multiplications involving for instance the input-hidden and hidden-hidden weight matrices in the LSTM equations [10] are replaced with convolutions. This enables the parameter sharing and location invariance that comes with convolutional layers, while maintaining a recurrent setting. This type of layer structure is not to be confused with the more common setup of stacking RNN layers on top of convolutional layers (as in the VGG+LSTM architecture), which differs in two important and related aspects from the C-LSTM. First, any input to a standard RNN layer has to be flattened to a one-dimensional vector beforehand, which shatters the spatial grid patterns of an image. Second, by extracting the spatial features separately, hence by performing down-sampling convolutions of the images before any sequential processing, one risks losing important spatio-temporal features. In effect, there are features of a video clip that do not stem exclusively from either spatial or temporal data, but rather from both. For this reason, C-LSTM layers are especially suited to use for video analysis.

Next, we present the details of the C-LSTM networks in one and two streams from our experiments.

Convolutional LSTM in one and two streams. The one-stream C-LSTM model (C-LSTM-1) has four stacked layers of 32 C-LSTM units each with max pooling and batch normalization between every layer, followed by a dense layer and sigmoid output.

The two-stream C-LSTM (C-LSTM-2) is shown in Figure 2. This model consists of two parallel C-LSTM-1 streams. The idea, just like in [30], is for the motion stream (optical flow) to complete the spatial stream (RGB). However, our C-LSTM two-stream model differs in some aspects from the original architecture presented in [30]. To begin with, in [30], only the motion stream takes input with a temporal span while the spatial one processes single (momentaneous) RGB frames. Furthermore, the motion stream input, consisting of k concatenated optical flow frames, is not treated recurrently but as one single input volume. In [30], both streams are feedforward CNNs, whereas in our case both streams are recurrent. In addition, we choose to fuse our two streams at the feature-level, in contrast to [30] who fuse the class scores only after the softmax output layer. In particular, we fuse the two tensor outputs from the fourth layers in both streams by either element-wise multiplication or addition (Table 3), in order to use the optical flow features as an attention mask on the RGB features. To this end, the optical flow is computed using rather large averaging windows, which results in soft, blurry motion patterns, similar to the attention results presented in [28] (Figure 1). When using multiplication, the attention mask behaves like an AND gate; when using addition, it behaves like the softer OR gate. Both types of fusion emphasize the parts of the image where pixels have moved, which are likely to be of interest to the model.

4.2. Implementation details

The top dense and LSTM layers of the VGG+LSTM model are trained from scratch and the convolutional layers of the VGG-16 base model are pre-trained on ImageNet [4] as an initialization. Its LSTM layer has 512 units, which hyperparameter was set after trials with [32, 256, 512, 1024] units. InceptionV3 is trained from scratch since this worked better than with ImageNet pre-trained weights. The top part of the original model is replaced with a dense layer with 512 units and global average pooling before the output layer. All architectures have a dense output layer with sigmoid activation and binary cross-entropy as loss function. The respective optimizers used are shown in Table 2. For the VGG+LSTM, we also trained with the Adam optimizer, as in [25], but Adadelta gave better results. We use larger frames when training the InceptionV3 models than when training the sequential models (Table 2) because InceptionV3 has a minimum input size. When training on sequences, they consist of 10 frames each. The sequences are extracted back-to-back without overlap (stride 10). We use early stopping of 15 epochs on a maximum of 100 epochs throughout the experiments. All two-stream models use dropout with probability 0.2 after the fusion of the two streams. Since additive feature level fusing worked best for the C-LSTM-2, this is chosen as the method of fusion for the two-stream extensions of VGG+LSTM and InceptionV3 as well. Data augmentation for the sequences consists of horizontal flipping, random cropping and shading by adding Gaussian noise. The transformations are consistent across a given sequence (i.e. for random cropping, the same crop is applied for the whole sequence). Remaining hyperparameters and the code for the experiments can be found at https://goo.gl/9TPYNk.

5. Experiments and results

5.1. Evaluation method

The classification task is evaluated using leave-one-subject-out cross testing, meaning that for the equine dataset we always train on four subjects, validate on one and test on the remaining subject. By separating the subjects used for training, validation and testing respectively, we enforce generalization to unseen subjects. Subject 5 is always used as validation set for early stopping, on account of its relatively even class balance (Table 1). When it is tested on, subject 1 is used for validation instead. The results presented are the average of the six cross-subject test rotations, which additionally were repeated five times. The standard deviations in Table 3 are the averages of the standard deviations across the test folds for every run, while the standard deviations presented in Table 4 are the per-subject variations across the five runs, which are smaller. F1-score is the harmonic mean of precision and recall. The reason to use F1-score in addition to accuracy is that it is a more cautious measure when a dataset contains class imbalance.

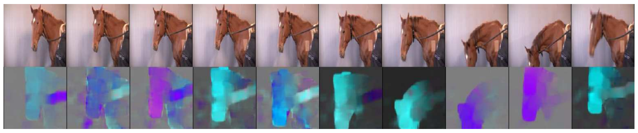


Figure 1: Example sequence from the dataset. Optical flow on second row. In the two-stream model, the optical flow serves as attention when fused with the RGB frames.

The labels are set globally according to the presence of pain induction from the recording protocol for the different video clips before frame extraction, meaning that frames belonging to the same clip always have the same ground truth. Nevertheless, the loss function is optimized per frame during training, which means that the network outputs one classification per frame. When computing the F1-score at evaluation time for sequential models, we want to measure their performance on the aggregate 10-frame sequence-level. To this end, a majority vote is taken of the classifications across the sequence. If there is a tie, the vote is random.

5.2. Veterinary expert baseline experiment

As a baseline comparison, four veterinarians with expert training in recognizing equine pain were engaged in classifying 51 five second-clips sampled at random from the dataset (but distributed across the subjects) as pain or no pain. Five seconds is the same temporal footprint that the sequential models are trained and tested on. The results in Table 3 show that it is not evident, even for an expert, to perform this kind of classification, highlighting the challenging nature of the task and dataset. The average F1-score (accuracy) of the experts was 54.6 (58.0%).

Since the clips are randomly extracted from the videos, it can happen that the horse looks away or interacts too much with an on-site observer. When the experts judged that it was not possible to assess pain from a certain clip, they did not respond. Thus, those results are not part of the averages. On average 5-10 clips out of the 51 were discarded by every rater. This ”cherry-picking” should speak to the advantage of the experts compared to the models, as well as the fact that the experts could re-watch the clips. Yet, on average, the experts perform 18.9 percentage points (p.p.)

Table 2: Overview of input details for the four evaluated models.

Model	Input, Equine data	Batch size	Sequential	Optimizer
C-LSTM-1	[10, 128, 128, 3]	16	Yes	Adadelta
C-LSTM-2	[10, 128, 128, 3] ×2	8	Yes	Adadelta
VGG+LSTM [25]	[10, 128, 128, 3]	16	Yes	Adadelta
VGG+LSTM-2 [25]	[10, 128, 128, 3] ×2	8	Yes	Adadelta
InceptionV3 [33]	[320, 240, 3]	100	No	RMSProp
InceptionV3-2 [33]	[320, 240, 3] ×2	50	No	RMSProp

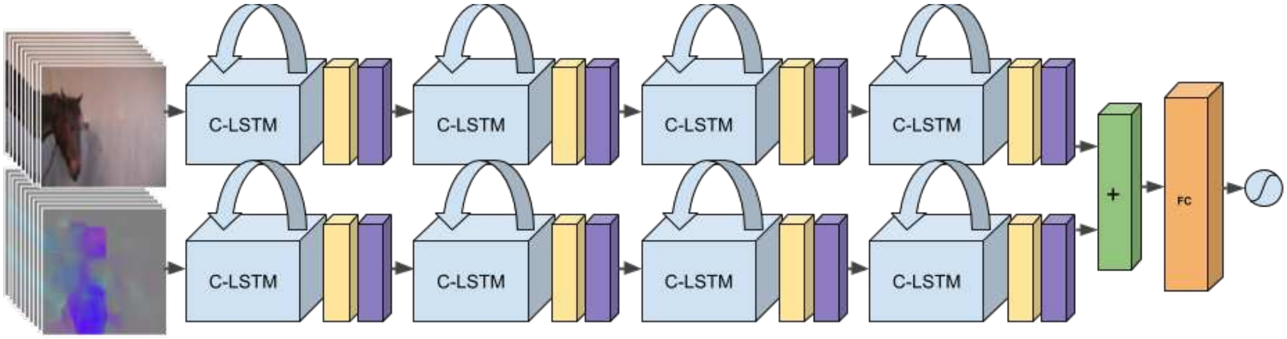


Figure 2: Our fully recurrent C-LSTM-2 model, which performed the best on the equine dataset. The yellow and purple layers are max pooling and batch normalization, respectively. The arrows on top of the C-LSTM layers symbolize recurrence.

worse than the C-LSTM-2 model.

Even so, the F1-scores varied among the raters. Their individual results were 52.7%, 50.7%, 72.2% and 42.7%. Interestingly, the rater that obtained 72.2% knew the dataset best out of the four and was present throughout the data collection. This rater's result, close to that of our best model, can thus be seen, to some extent, as testing on training data.

5.3. Automatic detection in the Equine Pain Dataset

Table 3 shows average results and standard deviations for the different evaluated models. The best obtained F1-score (accuracy) is 73.5 ± 18.3% (75.4 ± 14.1%) using the C-LSTM-2 architecture. Both C-LSTM-1 and C-LSTM-2 in their best settings outperform the relevant baselines.

The effect of two streams and temporal dynamics. We see a clear improvement of +2.2 p.p. from the C-LSTM one-stream model to its two-stream version, in terms of F1-score. For the VGG+LSTM model, there is no improvement from one to two streams when no data augmentation is used, and very little improvement (+0.1 p.p.) when data augmentation is used. The two-stream version of InceptionV3 in effect performs worse than its RGB one-stream version. When comparing the importance of the RGB and optical flow streams, respectively, for the three different models, we see that the flow alone performs better for the C-LSTM-1 (69.8% F1-score) than for the VGG+LSTM (66.1% F1-score) and the InceptionV3 (54.1% F1-score). Out of the three models, the performance gap between the modalities is the least for the C-LSTM-1. The above observations combined interestingly show that dynamics and sequentiality are more important to the C-LSTM model.

Furthermore, we see in Table 3 that the C-LSTM models perform better than the single-frame model InceptionV3, which can be considered as a baseline. Its best result is 68.1% F1-score for one stream with data augmentation. As a reminder, the input resolution to InceptionV3 is 320 × 240 pixels, compared to the 128 × 128 pixels for the C-LSTM networks. This underlines the better performance of the latter, even with nearly five times less pixels and orders of magnitude less parameters (Table 3).

The effect of data augmentation. We can observe in Table 3 that the C-LSTM and VGG+LSTM models trained without augmented data perform better than those trained with augmented data. For InceptionV3, the performance is better with data augmentation. A possible explanation for why the C-LSTM models do not learn the augmented data as well as the clean data is their small number of parameters, compared to the other models (Table 3). The fact that the C-LSTM models do not improve with data augmentation suggests that they are of the appropriate size for the dataset and have not overfit. The VGG-16 base model of VGG+LSTM has many parameters but we hypothesize that it is the single LSTM layer that might be too minimalistic for the augmented data in this case. In the future, when working on more varied datasets, it will likely become necessary to use a larger number of hidden units, in the interest of stronger network expressivity and generalizing ability. Nevertheless, it is reasonable to use a smaller number of hidden units for equine data than for example when training on a larger and more varied dataset such as UCF-101 [32]. For our applications, the data will always reside on a smaller manifold than general activity recognition datasets since we restrict ourselves to horses in a box or hospital setting. Data augmentation was only applied to the C-LSTM-2 (add) model since it performed better than the one with multiplicative fusion. In addition, the computational cost in training time was considerable, especially with augmented data.

Performance difference between subjects. The performance of VGG+LSTM varies markedly between subjects (Table 4). On that account, it has the highest global standard deviation among the models. When inspecting its classification decisions for subjects 1 and 2, for which it often scored above 90% F1-score, the model had sometimes identified hay in the corner or the barred background as being significative of the pain category. The horses it performed the worst on, subjects 5 and 6, have screens as background in the dataset and thus not as many details to overfit to.

As shown in Table 4, the results are generally better for

Table 3: Results (% F1-score and accuracy) for binary pain classification per evaluated model. C-LSTM-2 (add) has the best average result. InceptionV3 is a single-frame model, while VGG+LSTM and C-LSTM are sequential models. Data augmentation was only applied to C-LSTM-2 (add), for computational reasons, since it performed better than the one with multiplicative fusion.

	No data augmentation		**Data augmentation**		
One-Stream Models	**Avg. F1**	**Avg. accuracy**	**Avg. F1**	**Avg. accuracy**	**# Parameters**
InceptionV3 [33], Flow	54.1 ± 10.9	60.8 ± 10.3	57.5 ± 12.3	60.4 ± 8.5	$22,852,898$
InceptionV3 [33], RGB	62.6 ± 19.4	66.5 ± 16.8	68.1 ± 15.8	68.8 ± 14.2	$22,852,898$
VGG+LSTM [25], Flow	65.3 ± 16.6	67.6 ± 13.6	50.5 ± 17.0	60.9 ± 13.2	$57,713,474$
VGG+LSTM [25], RGB	69.4 ± 28.8	72.8 ± 22.9	63.8 ± 26.9	70.3 ± 19.8	$57,713,474$
C-LSTM-1, Flow (Ours)	69.8 ± 10.3	70.8 ± 9.4	59.5 ± 11.6	64.1 ± 9.8	$731,522$
C-LSTM-1, RGB (Ours)	71.3 ± 19.4	73.5 ± 16.3	64.0 ± 20.5	68.7 ± 14.9	$731,522$
Two-Stream Models					
InceptionV3-2 (add)	62.4 ± 20.5	66.3 ± 16.7	55.4 ± 18.0	59.7 ± 15.3	$45,704,770$
VGG+LSTM-2 (add)	69.4 ± 29.7	74.6 ± 22.5	63.9 ± 26.4	70.5 ± 18.9	$115,425,922$
C-LSTM-2 (mult.) (Ours)	67.3 ± 19.1	72.2 ± 14.9	-	-	$1,458,946$
C-LSTM-2 (add) (Ours)	$\mathbf{73.5 \pm 18.3}$	75.4 ± 14.1	66.5 ± 20.6	68.8 ± 13.8	$1,458,946$
Veterinary expert	54.6 ± 11.0	58.0 ± 13.6	n/a	n/a	n/a

subjects 1-5 compared to subject 6. Had we not included subject 6 in our final score, our best average would instead have been $80.9 \pm 8.6\%$ F1-score. This horse displayed atypical behavior consisting of extreme playfulness and interaction with the human observer, because it had not been trained to stand still like the other subjects (Section 3.2).

The accuracies in Table 3 in general have lower standard deviations than the F1-scores. This is because the F1-score discredits results where the model exclusively chooses one class for a whole test round. This often happened when the classification failed, as was almost always the case for subject 6. The F1-scores for subject 6 are around 40%, whereas in terms of accuracy these results are instead around 50%, hence reducing the standard deviation gap.

Tuning of C-LSTM-2. For C-LSTM-2, the softer fusion by addition performs better than fusion by multiplication. The optical flow is computed at 16 fps and is matched to the corresponding 2 fps RGB frames. If the horse moves fast, there can be discrepancies when overlaying the RGB image with an AND mask (corresponding to multiplicative fusion) not fitting the shape of the horse. This could explain the better performance of the additive fusion.

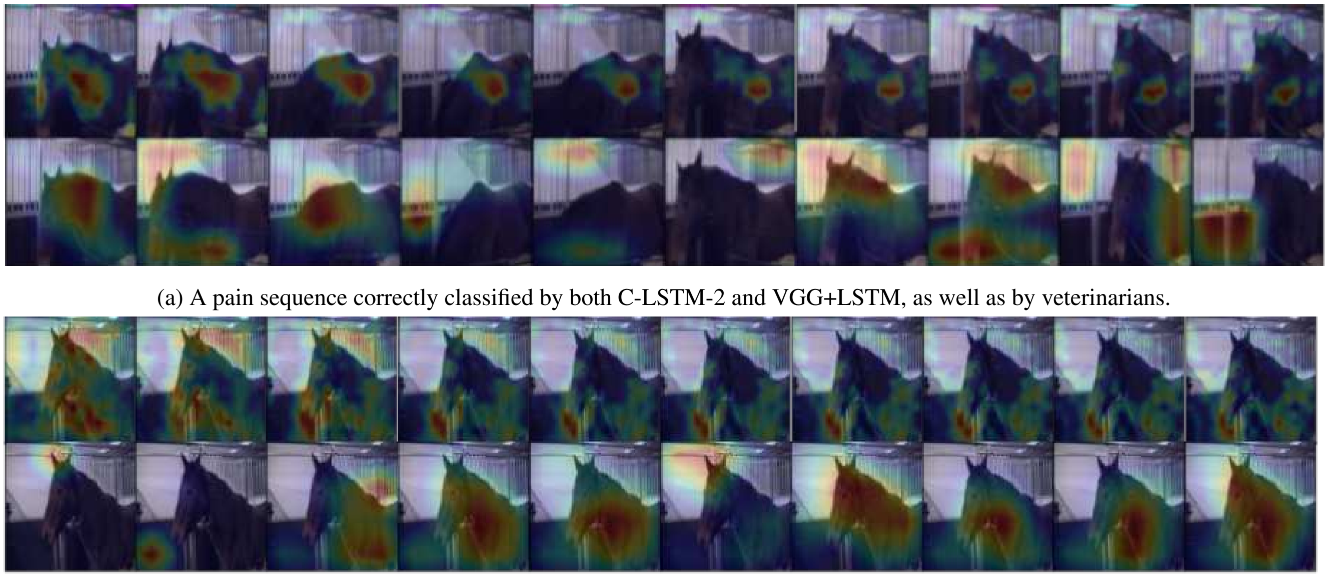
Comparison to Lu et al. To our knowledge, [16] is the only previous study done on automatic pain recognition in larger non-human species. An accuracy of 67% is obtained on a three-class pain level classification task (76.8% for two classes) via the detection of facial action units from images of sheep. The mapping from action units to pain level is done using the sheep pain facial expression scale, SPFES [20]. However, this result is not obtained on a subject-exclusive train/test split and the data is culled for frontal faces with visible action units. Importantly, the labels are set based on action units, and not on induced pain. The fact that the ground truth is based on visible action units makes the task easier. The authors also evaluate their method for unseen subjects; a confusion matrix is presented in [16] with results from a three-class task. We translate this into two classes, to be able to compare our results. The F1-score (accuracy) when testing on unseen subjects is 58.4% (58.1%), 15 p.p. less than our best result.

Table 4: Per-subject F1-scores for C-LSTM-2 and VGG+LSTM. The distribution is more even for C-LSTM-2.

Subject:	1	2	3	4	5	6
C-LSTM-2	87.1 ± 4.8	83.4 ± 5.6	79.7 ± 11.2	73.4 ± 5.8	76.5 ± 11.5	40.8 ± 3.6
VGG+LSTM	90.5 ± 4.1	94.4 ± 3.2	77.2 ± 16.3	89.1 ± 4.8	32.1 ± 7.0	33.8 ± 4.2

What do the models consider as pain? We compute saliency maps for the two best models, C-LSTM-2 and VGG+LSTM, to investigate their classification decisions. Two example sequences are shown in Figure 3. The saliency maps are made using the Grad-CAM method [27], taking the gradient of the class output with respect to the different filters in the last convolutional layer (for C-LSTM-2, we take the gradient with respect to the last layer in the RGB-stream). The filters are then weighted according to their corresponding gradient magnitude. The one with the highest magnitude is considered the most critical for the classification decision. This particular filter, upsampled to the true image size, is then visualized as a heatmap.

The sequence in Figure 3b shows a rather still horse with its head in an upright position. The sequence was misclassified as pain by all four experts but was correctly classified by the two computer models. The veterinarians based



(a) A pain sequence correctly classified by both C-LSTM-2 and VGG+LSTM, as well as by veterinarians.

(b) A non-pain sequence correctly classified by both C-LSTM-2 and VGG+LSTM, misclassified by all veterinarians.

Figure 3: Saliency maps for two sequences and two different models (C-LSTM-2 on the first row, VGG+LSTM on the second row in each figure). We observe that in general, the C-LSTM-2 model focuses on the horse and follows its movement, whereas the VGG+LSTM model has overfit (looks at the background) and does not follow a smooth temporal pattern across the sequence. In 3a, the C-LSTM-2 is attentive to the pose of the horses' shoulder and also zooms in on the tense eye area in the last three frames. In 3b, the muzzle, neck and eye area are in focus for the C-LSTM-2, whereas it appears that the VGG+LSTM has mostly learned to detect the presence of a horse.

their decision mostly on the tense triangular eye in the clip (sign of pain), visible in the proper resolution of the image, whereas the C-LSTM-2 seems to have mostly focused on the relaxed muzzle and upright head position (signs of non-pain). The VGG+LSTM seems to look at the entire horse, and frequently changes focus to the background and back again. Our impression is that the VGG+LSTM does not have a distinct enough focus to properly assess pain. Furthermore, Figure 3a, indicates that the VGG+LSTM model has overfit to the dataset, often looking at the background. The non-smooth temporal pattern of the saliency maps for VGG+LSTM in both Figures 3a and 3b could be explained by this model's separation of spatial and temporal feature extraction. By contrast, C-LSTM-2 tends to follow relevant parts of the horse in a temporally smooth fashion, which gives us more confidence in its classifications. This study is not conclusive but gives an indication to what kind of patterns the models seem to learn from the data, as well as a preliminary explanation to the better performance of C-LSTM-2. Grad-CAM was applied to the single-frame model InceptionV3 as well, but the results were uninformative and showed no particular saliency, while at the same time the model gave over-confident classification decisions.

6. Conclusions

From our comparisons of three main models and their respective two-stream extensions, varying in the extent by which they model temporal dynamics, it appears that the spatio-temporal unfolding of behavior is crucial for pain recognition in horses. In this work, we have found that sequential imagery and a model such as the Convolutional LSTM that simultaneously processes the spatial and temporal features improves results compared to models that process single frames. Sequentiality seems particularly important when assessing equine pain behavior, since horses do not convey a single straightforward face to show that they are in pain, the way humans can. Our obtained results on data labeled according to pain induction surpass a veterinary expert baseline and outperform the only previously reported result [16] on subject-exclusive larger animal pain detection by a clear margin, using no facial action unit annotations or pre-processing of the data.

6.1. Future work

In future work, we would like to systematically investigate what the model finds salient when it comes to equine pain behavior and compare it to today's veterinary pain scoring. Interpretability is becoming increasingly important in deep learning, as shown by works such as [7] and [27]. We see an interesting research direction for exploring the spatio-temporal patterns of equine pain both in terms of the learned representations and classification attribution. In the coming months, we will append more video labeled in terms of pain behavior to the database of [8].

References

[1] G. Bussieres, C. Jacques, O. Lainay, G. Beauchamp, A. Leblond, J. L. Cadore, L. M. Desmaizieres, S. G. Cuvelliez, and E. Troncy. Development of a Composite Orthopaedic Pain Scale in Horses. *Research in Veterinary Science*, 85, 2008.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[3] E. D. Costa, M. Minero, D. Lebelt, D. Stucke, E. Canali, and M. C. Leach. Development of the Horse Grimace Scale (HGS) as a Pain Assessment Tool in Horses Undergoing Routine Castration. *PLoS ONE*, 9(3), 2014.

[4] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] P. Ekman and W. V. Friesen. Measuring Facial Movement. *Environmental Psychology and Nonverbal Behavior*, 1, 1976.

[6] G. Farnebäck. Two-Frame Motion Estimation Based on Polynomial Expansion. *Lecture Notes in Computer Science*, 2749(1):363–370, 2003.

[7] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman. What have we learned from deep representations for action recognition? *CoRR*, abs/1801.01415, 2018.

[8] K. B. Gleerup, B. Forkman, C. Lindegaard, and P. H. Andersen. An equine pain face. *Veterinary Anaesthesia and Analgesia*, 42, 2015.

[9] K. B. Gleerup and C. Lindegaard. Recognition and Quantification of Pain in Horses: A Tutorial Review. *Equine Veterinary Education*, 28, 2016.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[11] S. E. Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. Recurrent Neural Networks for Emotion Recognition in Video. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI '15*, 2015.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-Scale Video Classification with Convolutional Neural Networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[13] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek. VideoLSTM convolves, attends and flows for action recognition. *CoRR*, abs/1607.01794, 2016.

[14] G. C. Littlewort, M. S. Bartlett, and K. Lee. Automatic Coding of Facial Expressions Displayed During Posed and Genuine Pain. *Image and Vision Computing*, 27, 2009.

[15] E. J. Love. Assessment and Management of Pain in Horses. *Equine Veterinary Education*, 21, 2009.

[16] Y. Lu, M. Mahmoud, and P. Robinson. Estimating Sheep Pain Level Using Facial Action Unit Detection. In *12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 2017.

[17] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews. Painful data: The UNBC-McMaster Shoulder Pain Expression Archive Database. In *Face and Gesture 2011*, pages 57–64, March 2011.

[18] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[19] S. McDonnell. Is it Psychological, Physical, or Both? In *Proceedings of the 51st Annual Convention of the American Association of Equine Practitioners, Seattle, Washington, USA*, 2005.

[20] K. M. McLennan, C. J. Rebelo, M. J. Corke, M. A. Holmes, M. C. Leach, and F. 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.

[21] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond Short Snippets : Deep Networks for Video Classification. 2015.

[22] R. Pascanu, T. Mikolov, and Y. Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012.

[23] J. Price, N. Clarke, E. Welsh, and N. Waran. Preliminary Evaluation of Subjective Scoring Systems for Assessment of Postoperative Pain in Horses. *Veterinary Anaesthesia and Analgesia*, 30, 2003.

[24] L. C. Pritchett, C. Ulibarri, M. C. Roberts, R. K. Schneider, and D. C. Sellon. Identification of Potential Physiological and Behavioral Indicators of Postoperative Pain in Horses After Exploratory Celiotomy for Colic. *Applied Animal Behavioural Science*, 80, 2003.

[25] P. Rodriguez, G. Cucurull, J. Gonalez, J. M. Gonfaus, K. Nasrollahi, T. B. Moeslund, and F. X. Roca. Deep Pain: Exploiting Long Short-Term Memory Networks for Facial Expression Classification. *IEEE Transactions on Cybernetics*, 2017.

[26] D. C. Sellon, M. C. Roberts, A. T. Blikslager, C. Ulibarri, and M. 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, 2004.

[27] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016.

[28] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.

[29] X. Shi, Z. Chen, and H. Wang. Convolutional LSTM Network : A Machine Learning Approach for Precipitation Nowcasting. *CoRR*, abs/1506.04214, 2015.

[30] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems 27*, 2014.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[32] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[33] C. Szegedy, V. Vanhoucke, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. *CoRR*, abs/1512.00567, 2014.

[34] A. H. Tuttle, M. J. Molinaro, J. F. Jethwa, S. G. Sotocinal, J. C. Prieto, M. A. Styner, J. S. Mogil, and M. J. Zylka. A deep neural network to assess spontaneous pain from mouse facial expressions. *Molecular Pain*, 14:1744806918763658, 2018. PMID: 29546805.

[35] J. P. van Loon and M. 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, 2015.

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9912 LNCS, 2016.

[37] J. Wathan, A. M. Burrows, B. M. Waller, and K. McComb. EquiFACS: The Equine Facial Action Coding System. *PLoS ONE*, 10(8), 2015.

[38] J. Zhou, X. Hong, F. Su, and G. Zhao. Recurrent Convolutional Neural Network Regression for Continuous Pain Intensity Estimation in Video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1535–1543, 2016.

[39] G. Zhu, L. Zhang, P. Shen, and J. Song. Multimodal Gesture Recognition Using 3-D Convolution and Convolutional LSTM. *IEEE Access*, 5:4517–4524, 2017.