Face detection and Grimace Scale Prediction of White Furred Mice

Andrea Vidal, Sumit Jha, Shayne Hassler, Theodore Price, Carlos Busso

Abstract

Studying the facial expressions of humans has been one of the major applications of computer vision. An open question is whether common machine learning techniques can also be used to track behaviors of animals, which is a less explored research problem. Since animals are not capable of verbal communication, computer vision solutions can provide valuable information to track the animal's state. We are particularly interested in pain neurobiology research, where rodent models are extensively used to investigate pain interventions. A grimace scale is used to understand the suffering of a mouse in the presence of interventions, which is inferred from various facial features such as the shape of the eyes and ears. In this work, we automate the prediction of the grimace scale on white furred mice using a machine learning approach, following the same principles used for human facial expression recognition: face detection, landmark region extraction, and expression recognition. We demonstrate the use of the you only look once (YOLO) framework for face detection of the mice with outstanding results. For eye region extraction and grimace pain prediction, we propose a novel structure based on a dilated convolutional network. The experimental results are promising, showing that it is possible to differentiate among the pain scale of the mice.

Keywords: Mice pain detection, Deep Learning, Convolutional Neural Networks

1. Introduction

Animals have been useful companions from the beginning of human civilization. Apart from emotional support as pets, animals have also enabled us to perform research in medicine and neuroscience by testing various procedures

before we use them on humans. By evaluating candidate medical procedures in animals, we can identify possible secondary effects leading to potential risks for human beings. Our research is particularly interested on pain neurobiology research aiming to identify interventions to reduce chronic pain in human patients. One important aspect in these procedures is to study the level of pain in animals. Existing approaches manually estimate the pain level by observing the behaviors of the animals. Having an automatic system for pain detection in animals can serve as an important tool to accelerate advances in these fields, reducing the time of medical testing procedures.

A white mouse is a prime example of an animal that has been used in neuroscience experiments. During these experiments, mice's behaviors are analyzed to estimate the nociception by observing "pain-like" cues on mice. The analysis of the mice's behaviors can be categorized into stimulus-evoke and non-stimulus evoke. Stimulus-evoke methods consist of using mechanical or thermal (heat or cold) stimuli, which change the mice's behaviors, followed by studying their reactions. Non-stimulus evoke methods measure the spontaneous or background pain. Because of the inability of the animals to communicate, the suffering of the animals is challenging to observe. Therefore, pain measurement can be estimated using several methods including burrowing tests, weight bearing, and grimace scales (Deuis et al., 2017). The grimace scale (Langford et al., 2010) captures the pain level of the mice by observing various characteristics of the facial features of the animal. The approach consists of observing the orbital tightening, ears, nose, cheeks and whiskers. The cues are used to define whether the animal does not suffer pain (value 0), feels moderate pain (or feels pain with uncertainty, value 1), or definitely suffers pain (value 2). However, manually performing these annotations is very time-consuming and unreliable. First, an experienced annotator has to identify clear frames with frontal views where the face of the mouse is visible. This is a time-consuming task since a mouse face is not clearly visible in most frames given the sudden movements of the mouse. However, the annotator has to identify the pain cues from the mouse's face. In this work, we propose a system that can automatically detect the level of pain in the

mice using solutions from machine learning and computer vision. For this purpose, we present a system based on deep learning capable of (1) detecting frontal faces of white furred mice, and (2) estimating the pain level using the eye cues of white furred mice.

We propose a system that predicts the grimace level in a mouse based on the orbital tightening. We use a database collected by the Brain and Behavioral Science Department at the University of Texas at Dallas. This database includes data from eight white furred mice, subjected to induced pain. The proposed system solves two important problems: face detection and pain estimation. The first part corresponds to a face detection system based on the *you only look once* (YOLO) network (Redmon & Farhadi, 2018), which helps us identify the frames with a clear frontal face, saving these frames for further analysis. The network shows high accuracy in predicting a bounding box for the mouse face with a mean *intersection over union* (IoU) score of 0.87. The second part corresponds to the grimace scale prediction, where we predict the grimace scale of the mouse from the orbital tightening by localizing the eye region in a mouse face image. We use the detected frontal face to obtain a region around the eyes of a white mouse using a dilated *convolutional neural network* (CNN). The eye patch is the input for our classification model that predicts the grimace scale in the mouse. The model reaches a high accuracy especially in predicting between no pain (value 0) and high pain (value 2), achieving an accuracy of 97.2%. While the accuracy drops when we consider the class "mild pain" (value 1) in a three-class problem, the results are still higher than the baselines. We compare our mouse grimace scale prediction model with the approach proposed by Tuttle et al. (2018) and two baselines based on neural network and *support vector machine* (SVM) trained with *histogram of gradient* (H0G) features. The results show that our proposed model clearly outperforms all the baseline models. The contributions of this study are:

- We collect the infrastructure and resources to leverage machine learning solutions in a novel problem of face detection and pain estimation of white furred mice

- We repurpose a YOLO network to perform face detection of white furred mice
- We propose a *deep neural network* (DNN) for pain detection of white furred mice

2. Related Work

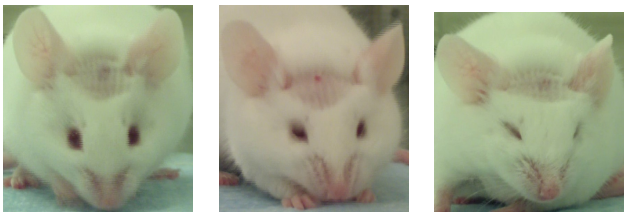
2.1. Face Analysis in Animals

Facial expression analysis is a widely studied research topic in humans. Over the years features such as facial *action units* (AU) (Tian et al., 2001; Zhao et al., 2016) and facial landmarks (Baltrušaitis et al., 2013; King, 2009) have been used to study the emotional state of a person. With the success of computer vision algorithms in detecting human facial expressions, the community has recently explored the idea of using similar tools to study animal behaviors. However, the progress on facial expression analysis in animals is still limited (Descovich et al.). Studies in this area have explored pain detection algorithms in animals based on facial landmarks (McLennan & Mahmoud, 2019; McLennan et al., 2016), the entire face (Tuttle et al., 2018), or parts of the face (Kopaczka et al., 2018). Descovich et al. discussed the use of landmarks on an animal face to identify different emotional states related to the wellbeing of the animal. They discussed various emerging initiatives in the field of animal behavior analysis using computer vision solutions. Similarly, Burghardt & Ćalić (2006) proposed methods similar to human tracking to study the locomotive behaviors of animals in the wild. They used this method to track the movement of lions and classify them into multiple semantic categories such as stalking, standing and trotting. McLennan et al. (2016) created a tool defined as the *sheep pain facial expression scale* (SPFES) to determine the pain of sheep using action units. Subsequently, McLennan & Mahmoud (2019) designed a system with this tool to extract features from relevant regions of the face, such as eyes, nose, and mouth using *histogram of oriented gradients* (HoG). They estimated the action units using a *support vector machine* (SVM) trained with these HoG features. Lu et al. (2017) also proposed predicting the pain level of sheep by training a SVM with HoG features on localized parts of the face such as the nose, ears, and eyes. Hewitt & Mahmoud (2019)

proposed a pose aware landmark localization approach in animals following similar approaches used in head pose estimation and landmark localizations in humans. They also introduced a sheep database with 850 facial images annotated with 25 facial landmarks. Van Loon & Van Dierendonck (2018) provided a survey in the field of pain detection in horses, describing two main types of pain scales. The first one is singular and composite pain scales. The second is the facial expression-based pain scales, which is based on action units that result from contractions of certain facial muscles when the horse is in pain. The scale allows a veterinarian to determine the type of pain that the horse is suffering.

2.2. Mouse Grimace Scale

White furred mice are one of the most common animals used in laboratory experiments. They are widely used in testing drugs for diabetes, cancer and chronic pain. A major factor when performing experiments with animals is the ability to quickly and efficiently detect when the animal is suffering. Langford et al. (2010) first introduced a facial action coding system for mice referred to as the *mouse grimace scale* (MGS). The grimace scale in mice depends on five features from the face: orbital tightening, nose bulge, cheek bulge, ear position and whisker change. These facial action units take the values 0 (no pain), 1 (mild pain or pain probably present) and 2 (high pain or pain definitely present). Figure 1 shows examples of this scale on mice. In our study, we design a framework to detect the grimace scale of mice from the orbital tightening using automatic computer vision algorithms.



(a) Grimace level 0 (b) Grimace level 1 (c) Grimace level 2

Figure 1: Examples of frames from the database with their grimace label.

2.3. Relation to Prior Work

The closest studies to our paper are the studies of Kopaczka et al. (2018), and Tuttle et al. (2018). Kopaczka et al. (2018) proposed an eye detection method based on deep learning using CNNs to detect the eye region for black mice. The mice were recorded using a red background to simplify the task. Tuttle et al. (2018) used deep learning for the binary detection of pain in white furred mice (e.g., pain versus no pain). They used an end-to-end framework based on the Inception V3 architecture (Szegedy et al., 2016). They tested their model with images of mice during a laparatomy surgery and obtained high confidence in pain detection, which was comparable to human annotators.

We focus our work on two key aspects of automatic detection of the grimace scale. The first contribution focuses on obtaining key frames with clear frontal faces from continuous video that can be helpful in effectively estimating the pain levels of mice. This is an important task because the manual identification of clear frames is time-consuming. The previous work of Kopaczka et al. (2018) proposed a method for detecting clear frames based on eye segmentation. Unlike that work, we propose the use of a pretrained neural network on the field of object recognition and adapt it to mouse face detection obtaining high accuracy. The second contribution focuses on using these key frames to detect the grimace scale based on orbital tightening. In contrast with the work of Tuttle et al. (2018) that predicts the presence of pain or no-pain on the mouse, we predict the pain level on the grimace scale (0, 1, and 2). This is a difficult task, because we are not only predicting if the mouse is in pain, but also predicting the pain intensity, resulting is an F1-score of 0.718.

3. Database

This section discusses the database used in our study. The database contains videos of eight different white furred mice collected in multiple sessions across different days (*Institute of Cancer Research* (ICR) mouse strain). The mice are part of a neurobiology research study aiming to identify interventions to reduce chronic pain conducted by the Brain and Behavioral Science Department at the

University of Texas at Dallas (UT Dallas). The mouse procedures were approved by the *Institutional Animal Care and Use Committee* (IACUC) at UT Dallas. The mice were administered compound 48/80 onto the outermost membrane enveloping the brain and spinal cord (dura mater). The injection of this component is performed using internal cannulas, where the projection is determined individually for each mouse so that there is no damage to the dura. The supradural administration of the component was placed on the junction of the sagittal and lambdoidal sutures. Once the component is injected each mouse recovers and returns to its individual home cage. Hassler et al. (2019) discussed the protocol in detail. The behaviors of these mice are recorded using high resolution cameras (Samsung HMX-QF20 Full HD 1920 × 1080). The cameras are placed at the front of the cage to make sure each camera only records data from a single mouse.

We manually selected 2,222 frames from these videos to be annotated for the face detection task. The frames contain clear frontal view of the mouse face. We annotate bounding boxes around the mouse face, creating the resource to train automatic algorithms for face detection. We use data from five animals for the train set (1,361 frames), one animal for the development set (453 frames), and two animals for the test set (408 frames). We refer to this part of the dataset as *Sub-dataset 1*.

We also create annotations for the pain estimation task using eight mice. Images of frontal faces of the animals are used to annotate the MGS of the mice. Each image is annotated with a value for orbital tightening, nose bulge, cheek bulge, ear position and whisker change. The annotations are performed by experts, who rated the pain level as 0, 1 or 2 for each facial cue, following the scale discussed in Section 2.2. We annotate a total of 1,087 images. We refer to this subset as *Sub-dataset 2*. Since, our experiment focuses on the grimace scale prediction from the orbital tightening, we annotate each image with the bounding box around the eyes, so that we can train models that give high attention to the eye region. For experiments related to grimace prediction, we follow a leave-one-out cross-validation scheme. We select data from one animal as the test set. Of the remaining seven animals, data from one are randomly chosen to be used as the

development set. Data from the remaining six mice are used for training. We repeat this procedure eight times, each time using a different animal for the test set. The results reported in this study are compiled using all the frames in the corpus, aggregating the results across folds.

4. Proposed approach

We present a framework to detect the mouse face and predict its pain level (grimace scale). Both tasks are important in neuroscience research dealing with rodent models. Tracking a mouse in a continuous video is more challenging than tracking humans, because the behavior of the mice is more erratic and they tend to move around faster, especially when they are distressed. Mice do not look at the camera most of the time. Therefore, most frames either are blurred or contain non-frontal faces. Researchers have to manually identify frontal views of the mice to annotate visible cues (e.g., orbital tightening, nose bulge, cheek bulge, ear position and whisker change). Building a machine learning solution to automatize this process can greatly facilitate the work of neuroscientists, reducing the cost and effort for their study. Face detection is also a pre-processing step for our pain detection, since we only consider frontal views in our model. Likewise, having a solution for pain detection can be instrumental for longitudinal chronic pain analysis. The algorithm can be used to track frame-by-frame the pain level in the mice, creating data to study the temporal response associated with pain treatments.

4.1. Face detection

Face detection is the first step that needs to be applied before any mouse face analysis. Hence, our first task is to detect frames that provide a stable frontal face of the mouse. Face detection can help get rid of any background pixels that might confuse the model. Dalvi et al. (2021) presented several methods developed over the years for human face detection such as Viola-Jones algorithm, Haar-cascade classifier, and *Multi-task Cascaded Convolutional Networks* (MTCNN). The Viola-Jones algorithm (Viola & Jones, 2001) was the standard algorithm for object

detection applications. In recent times, CNN based techniques such as *regions with CNN features* (R-CNN) (Girshick et al., 2014), *single shot multiBox detector* (SSD) (Liu et al., 2016) and *you only look once* (YOLO) (Redmon et al., 2016) have improved performance, providing more efficient methods to perform object detection. The YOLO architecture has been successfully used to detect human faces (Batista et al., 2019; Chen et al., 2021, Garg et al., 2018). Therefore, this study repurposes the YOLO architecture for mice face detection.

YOLO is a deep neural network capable of detecting several objects, people, and animals in one frame independent of the size of the object. We use a pretrained version of YOLO v3 (Redmon & Farhadi, 2018) trained on the Open Images dataset (Krasin et al., 2017). A common approach with the YOLO architecture is to fine-tune the pretrained model to the mainstream task (Zheng & Amen, 2021; Zheng 2022). We following this approach, adapting the pretrained YOLO model to predict mouse faces. For this purpose, we only modify the output layer to predict a single class associated with the frontal faces of mice (i.e., the output layer consists of a single node). This modified YOLO model is trained for 100 epochs on our corpus. For the first 50 epochs, the entire model is frozen with the exception of the output layer. Then, we unfreeze all the parameters in the model, training the model for another 50 epochs.

4.2. Grimace Detection

The second goal of our study is to use key frames showing the mouse to predict the grimace level. We use the level of orbital tightening to estimate the grimace level. We focus on the eyes, since other cues used to annotate pain, such as whiskers, are less visible on regular cameras. Figure 2 gives an overview of the scheme used in our experiment. First, we generate a mask to highlight the eye region in the image using a dilated convolutional network. Subsequently, we use a neural network to perform classification to predict the grimace level

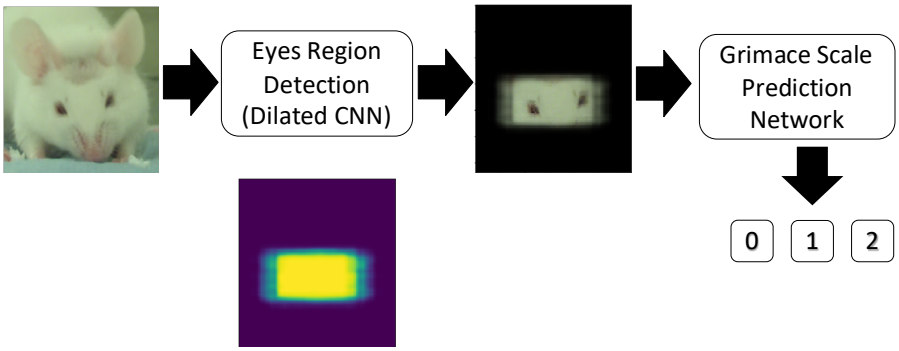


Figure 2: Scheme of our two-stage approach to predict grimace scale in mice.

from the masked image.

Dilated convolutional neural networks were proposed by Yu & Koltun (2016) as effective tools to perform semantic segmentation. This model can help increase the receptive field without loss of resolution. For the purpose of this work, the dilated convolutional neural network framework suits our problem of predicting the mice's eye region. The eye region detection is performed using the network architecture shown in Table 1. To train this model, we use the annotated bounding box for the eye region of the image creating a binary mask with 1 for the eye part, and 0 for the rest of the pixels. The trained model returns a mask with values from 0 to 1, providing a confidence score of how likely each pixel belongs to the eye region. By multiplying the resulting mask with the original image, we obtain the region of the mouse's eyes. Figure 2 shows an example of the eye patch. We use the eye region images for training the grimace scale prediction network to assess the level of pain (0, 1, or 2) following the MGS. Table 2 shows the architecture of the grimace scale prediction network, which consists of five blocks of 2D convolutional networks, each of them implemented with a kernel size of (3 × 3) and 16 channels. We use *rectified linear unit* (ReLU) activation followed by batch normalization and max pooling layers. The output of the final convolution layer is flattened and connected to a fully connected layer of size 32 followed by the output layer. The dimension of the output layer depends on the number of predicted classes. We explore two problems: a binary classification (0 versus 2),

and a 3-class problem (0, 1, and 2). We use softmax as the final activation layer to obtain scores that sum up to 1.

Table 1: Deep learning architecture for eye region detection using dilated CNN.

Layer	Spec	Dilatation	Activation	Dropout
Conv2D	**3, 3x3**	**0x0**	**ReLU**	**-**
Conv2D	**3, 3x3**	**0x0**	**ReLU**	**-**
Conv2D	**3, 3x3**	**0x0**	**ReLU**	**0.5**
Conv2D	**3, 3x3**	**2x2**	**ReLU**	**-**
Conv2D	**3, 3x3**	**2x2**	**ReLU**	**-**
Conv2D	**3, 3x3**	**2x2**	**ReLU**	**0.5**
Conv2D	**3, 3x3**	**4x4**	**ReLU**	**-**
Conv2D	**3, 3x3**	**4x4**	**ReLU**	**-**
Conv2D	**3, 3x3**	**4x4**	**ReLU**	**0.5**
Conv2D	**3, 3x3**	**8x8**	**ReLU**	**-**
Conv2D	**3, 3x3**	**8x8**	**ReLU**	**-**
Conv2D	**3, 3x3**	**8x8**	**ReLU**	**0.5**
Conv2D	**3, 3x3**	**16x16**	**ReLU**	**-**
Conv2D	**3, 3x3**	**16x16**	**ReLU**	**-**
Conv2D	**3, 3x3**	**16x16**	**ReLU**	**0.5**
Conv2D	**3, 3x3**	**32x32**	**ReLU**	**-**
Conv2D	**3, 3x3**	**32x32**	**ReLU**	**0.5**
Conv2D	**3, 3x3**	**-**	**ReLU**	**-**
Conv2D	**1, 3x3**	**-**	**Sigmoid**	**0.5**

Table 2: Deep learning architecture for Grimace scale prediction.

Layer	Spec	Activation	Dropout
Conv2D	16, 3x3	ReLU	-
Batch Norm	-	-	-
Max Pooling	2x2	-	-
Conv2D	16, 3x3	ReLU	-
Batch Norm	-	-	-
Max Pooling	2x2	-	-
Conv2D	16, 3x3	ReLU	-
Batch Norm	-	-	-
Max Pooling	2x2	-	-
Conv2D	16, 3x3	ReLU	-
Batch Norm	-	-	-
Max Pooling	2x2	-	-
Conv2D	16, 3x3	ReLU	-
Batch Norm	-	-	-
Max Pooling	2x2	-	-
Flatten	-	-	-
Dense	32	ReLU	0.5
Batch Norm	-	-	-
Dense	# classes	Softmax	-

5. Experimental Results

In this section, we present and discuss the results for mouse face detection and grimace scale prediction, which are the two tasks addressed in our study. All the deep learning models are implemented in Python using Keras (Chollet, 2017) with Tensorflow (Abadi et al., 2016) as backend.

Table 3: Number of true positive, false positive and false negative detections for the proposed and baseline models.

Method	True Positives	False Positives	False Negatives
Haar Cascade	122	286	1065
CNN-based baseline	265	140	143
Proposed YOLO-based model	408	0	0

5.1. Face detection results

As described in Section 4.1, we tune a pre-trained YOLO network with the face annotations of mice for 100 epochs. We use the eight animals from our *Sub-dataset 1* using the partitions described in Section 3. We compare our proposed YOLO-based method with a Viola-Jones object detection method trained with Haar features. We refer to this baseline as the Haar Cascade model. Also, we compare our proposed model with the method presented by Kopaczka et al. (2018), which is a CNN-based model for image segmentation. We adapt this model to detect the mouse's face. We refer to this baseline as the CNN-based model. We quantify the results of the face detection algorithms using the *intersection over union* (IoU), precision, and recall metrics. The IoU between two regions is a ratio between the area of the overlap between the two regions and the area covered by the union of the regions. We use the IoU scores to compute *true positive* (TP), *false positive* (FP), and *false negative* (FN) rates. We set an IoU threshold of 0.5, making the following rules: For each image, if at least one of the predictions have an IoU greater than 0.5, we define this prediction as a true positive; if there are no predictions in an image with IoU greater than 0.5, this image data has a false negative. All the remaining predictions in the image are classified as false positive. With these rules, we compute precision and recall metrics as follows,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

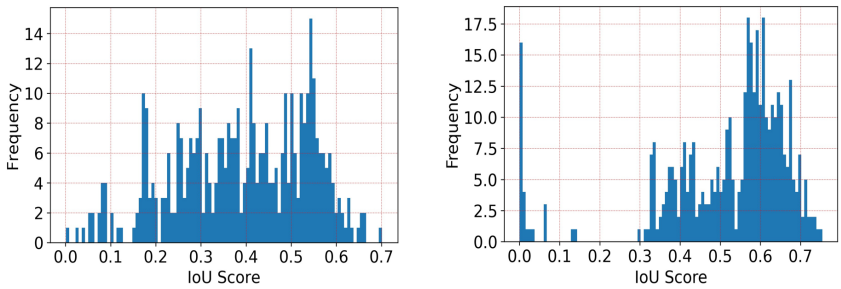
Table 4: Face detection performance for the proposed and baseline models, measured with the mean IoU, and precision and recall rates.

Method	mean IoU	Precision	Recall
Haar Cascade	0.39	0.103	0.299
CNN-based baseline	0.513	0.654	0.649
Proposed YOLO-based model	0.87	1.00	1.00

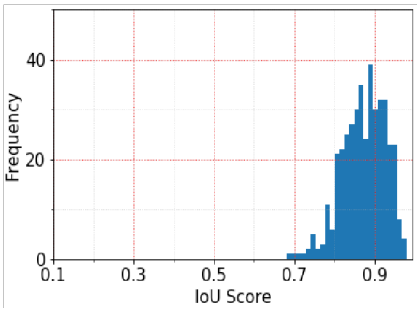
$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Table 3 shows the TP, FP and FN rates for the proposed YOLO-based method, the Haar Cascade baseline and CNN-based baseline. The number of false positives for the baseline models is high, and the number of false negatives is even higher. The direct impact of these results is shown in Table 4, which indicates that the Haar Cascade and CNN-based baselines obtain lower precision and recall rates than the proposed YOLO-based approach. Hence, these predictions are not reliable to use for grimace scale prediction. In contrast, the YOLO-based model yields a very high performance. The model is able to detect the face of the mice without any FP or FN case. Our recall and precision rates are 100%.

Figure 3 shows the histograms of the IoU scores in the test set using the proposed YOLO-based method, comparing the results with the Haar Cascade and CNN-based baselines. The baseline methods have a high number of false negatives with zero IoU (Table 3), which do not overlap with the true bounding boxes (Figure 4(a)). For the CNN-based baseline (Figure 4(b)), we observe that the region of the face is not completely detected, which can explain the high number of false positives. This method also detects some artifacts. Therefore, we show the histogram only considering the non-zero IoUs, where the prediction and the true bounding box had some overlap. This simplification does not affect our YOLO-based approach, since we do not have false negatives. For the Haar Cascade baseline, we observe many predictions with small overlaps with the true bounding box. There are no predictions with an overlap higher than 0.7. Hence,



(a) Haar Cascade baseline (b) CNN-based baseline

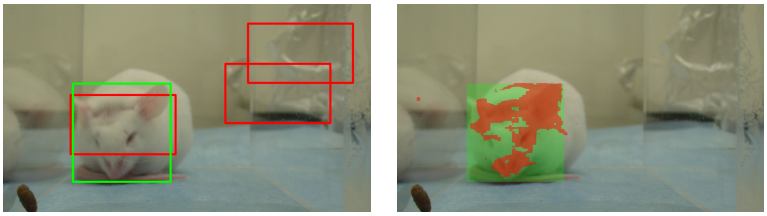


(c) Proposed YOLO-based Approach

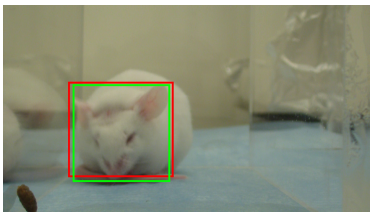
Figure 3: Histogram of IoU scores for mouse face detection using the proposed YOLO-based method, and the Haar cascade baseline.

the model has a very low recall score. For the CNN-based baseline, there is no IoU score higher than 0.76, which explains the low precision and recall rates. In contrast, most of the IoU scores are higher than 0.7 for our YOLO-based approach, indicating that our predictions can accurately detect the true mouse face. The mean IoU value is 0.87 with a standard deviation of 0.05. Figure 4(c) shows an example of the estimated face of the mouse compared with the ground truth. We observe high accuracy with high confidence in the estimation. We also observe that the detection is highly robust to pose variation and occlusion. Figure 5 shows various frames from a video, demonstrating that the YOLO-based algorithm efficiently detects frontal faces from a mouse even in challenging situations. These results show that the model can be used for extracting key frames from continuous videos, which can then be used to either automatically detect or

annotate the pain level in mice. The reliable face detection results enable us to predict the grimace scale relying on this automatic face segmentation algorithm.



(a) Haar Cascade baseline (b) CNN-based baseline



(c) Proposed YOLO-based Approach

Figure 4: Example of face detection performance using the proposed YOLO-based method, and the Haar cascade baseline. Green and red boxes represent the ground truth and the predicted boxes, respectively.

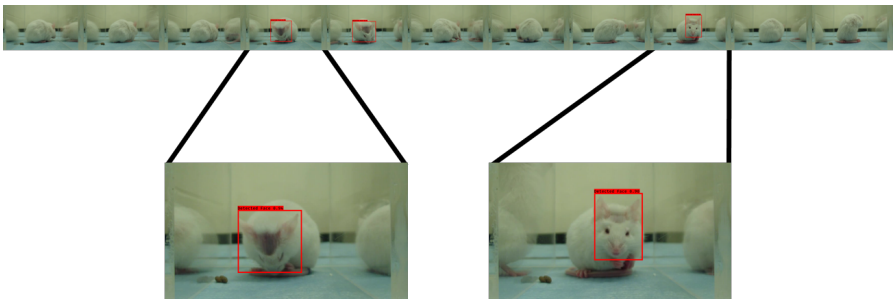


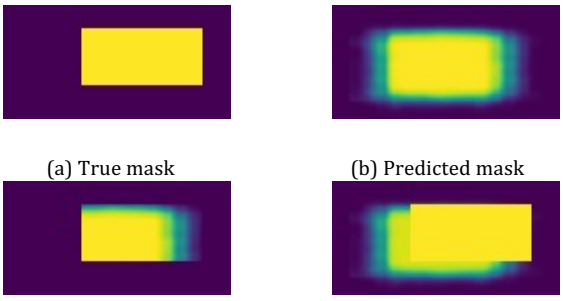
Figure 5: Performance of the YOLO-based approach over video frames. The approach can be useful in extracting key frames from the video with clear frontal faces.

5.2. Grimace scale prediction results

This section discusses the performance in pain detection using the orbital tightening. The first stage of this task is the detection of the eye region. For this purpose, we train the dilated CNN described in Table 1 for 1,000 epochs using

Adam (Kingma & Ba, 2015) as an optimizer with a learning rate 10^{-4}, and the *binary cross-entropy* as the loss function. Since our model predicts a soft region, we use an approximation of the IoU to quantify the difference between the ground-truth bounding box and the predicted area. This approximation is calculated as:

$$IoU = \frac{\Sigma(pred \cdot true)}{\Sigma(true + pred - pred \cdot true)} \tag{3}$$



(a) True mask (b) Predicted mask

(c) Intersection of masks (d) Union of masks

Figure 6: Illustration of the IoU calculation. IoU score for this example is 0.48 using the approximation in Equation 3.

This metric returns a number between 0 and 1 and provides a good idea about the overlap between predictions and the true bounding boxes. Figure 6 shows an example of the IoU calculation, exemplifying the true mask, predicted mask, intersection of masks and the union of masks. For that example, the IoU is 0.48. Figure 7 shows the histogram of IoU occurrence for all the data from the test dataset with annotated mouse grimace level. The mean value is 0.67. We observe that most values have an IoU score of 0.55 or higher, which means that our eye detection method is capable of capturing a large part of the eye region. This patch is then used to predict the grimace scale. Figure 8 shows three cases of predicted masks compared to the ground truth bounding box.

The second stage consists of using the eye patch to predict the grimace scale. We use the neural network described in Section 4.2 (Table 2). The network takes a square image with the masked eyes as input and provides a value for the grimace

scale. We learn a classification model with one-hot encoded labels. The model uses the AdaDelta optimizer (Zeiler, 2012) with a learning rate of 1.0. We compare

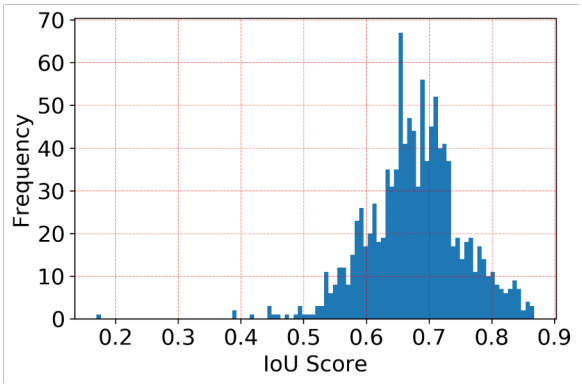
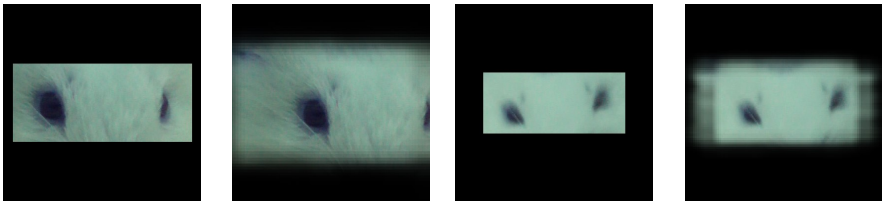
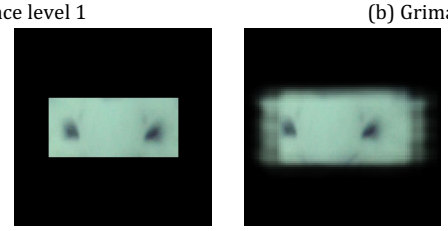


Figure 7: Histogram of IoU scores for eye region detection using the proposed dilated CNN.



(a) Grimace level 1 (b) Grimace level 1



(c) Grimace level 2

Figure 8: Three frame pictures from the test set with their corresponding ground-truth (squared bounding boxes) and our proposed method for eye region detection.

our proposed model with three baselines. The first baseline is a SVM model trained with HoG features, referred to as HOG-SVM. The second baseline is a neural network trained with HoG features with two hidden layers implemented with ReLU, and an output layer implemented with a softmax function. We refer to this baseline as HOG-NN. The third baseline is our implementation of the method

presented by Tuttle et al. (2018), which takes as input the entire image of the mouse's face to predict the grimace score. To study the influence of our eye region detection algorithm on the final grimace scale prediction, we train our grimace prediction model on both the *groundtruth* (GT) bounding box and the mask learned using our eye region detection algorithm. We add "GT" to the name of the models for the condition with manually annotated eye patches. We evaluate the proposed model and the first two baselines using the detected eye region patch and GT bounding box. The third baseline uses the entire face of the mouse, so we only use the true bounding box with the face region.

The grimace scale prediction networks are trained in two stages, following the idea of curriculum learning, which starts by training the network with easier samples, introducing harder samples in later epochs (Bengio, 2009). Since the samples with grimace level of 1 are more ambiguous, we start by training a binary classifier that recognizes no pain (value 0) versus pain (value 2). We train the binary model for 1,000 epochs using the *binary cross-entropy* loss function. Subsequently, we add the samples with label 1, adapting the output layer for three classes using the *categorical cross-entropy* loss function. We train the model for 500 more epochs. For consistency, we follow the same training procedure for the HOG-NN baseline and for the approach proposed by Tuttle et al. (2018). For the HOG-SVM, we separately train binary class and a multi-class classifiers.

Tables 5 and 6 show the weighted accuracy and the weighted F1-score for the predicted labels for the two and three class problems. The tables report the performance for our proposed method and the baselines. Our approach achieves a weighted F1-score of 0.976 for the two-class problem, which is an impressive result. The prediction of the grimace scale for the three-class problem also presents competitive results, achieving a weighted F1-score of 0.718. This result is promising, given the similarity in eye appearance between grimace level 0 and 1. We observe that deep learning methods obtain better results than the SVM method for both classification tasks. Between the deep learning methods, we observe that our proposed method outperforms the HOG-NN method and the approach proposed by Tuttle et al. (2018) for both classification problems. An

interesting result is that our approach is more robust to the use of predicted eye regions. Missed or partial detections of the eyes clearly affect the performance.

Table 5: Grimace scale prediction for the two-class problem. We report results using the predicted and GT bounding boxes.

	Weighted Accuracy	Weighted F1
Our method	0.972	0.976
HOG-NN	0.836	0.893
HOG-SVM	0.832	0.890
Tuttle et al. (2018)	0.848	0.895
Our method GT	0.963	0.974
HOG-NN GT	0.928	0.950
HOG-SVM GT	0.901	0.916

Table 6: Grimace scale prediction for the three-class problem. We report results using the predicted and GT bounding boxes

	Weighted Accuracy	Weighted F1
Our method	0.650	0.718
HOG-NN	0.526	0.604
HOG-SVM	0.464	0.527
Tuttle et al. (2018)	0.632	0.637
Our method GT	0.693	0.740
HOG-NN GT	0.621	0.677
HOG-SVM GT	0.588	0.646

For example, Figure 8(a) shows an example, where we barely detect the left eye of the mouse. The label for this case is 1. The proposed model predicts the value 0 when using the predicted eye region. However, the model predicts the right class when using the ground truth bounding box. However, the global results

when using either the GT bounding boxes or the predicted eye regions are very similar in spite of the eye detection errors. For the two-class problem, the differences in weighted accuracy and weighted F1-score are 1.27% and 0.16%, respectively. For the three-class problem, the differences in weighted accuracy and weighted F1-score are 4.24% and 2.13%, respectively. The differences for the baseline methods are much higher reaching differences up to 12.5%.

Tables 7 and 8 show the confusion matrices for the experiments performed using the predicted eye patches and GT bounding boxes using the proposed neural network. The left sides of the tables show the confusion matrices of the pain prediction for two classes. We observe that most of the samples are well classified. Out of 745 frames, only 18 (predicted eye patches) and 19 (GT bounding boxes) samples were not correctly predicted. The right sides of the aforementioned tables show the confusion matrices of the pain prediction for three classes. This is a more difficult task, where most of the errors are between classes 0 and 1, and between classes 1 and 2. The differences are difficult to distinguish even for experienced human annotators. For example, the cases in Figures 8(b) and 8(c) are predicted with grimace labels 2 and 1, respectively. However, their corresponding ground-truth are 1 and 2. Visual inspection of these frames shows that both images are very similar.

Table 7: Confusion matrices of the grimace detection model using the predicted bounding boxes for the eyes.

Predicted labels

True labels	0	2
0	539	13
2	5	188

Predicted labels

True labels	0	1	2
0	503	48	1
1	88	196	58
2	9	94	90

Table 8: Confusion matrices of the grimace detection model using the ground truth bounding boxes for the eyes.

True labels \ Predicted labels	0	2
0	545	7
2	12	181

True labels \ Predicted labels	0	1	2
0	485	67	0
1	48	194	100
2	0	71	122

6. Conclusions and Future Work

This work presented machine learning solutions to automatically detect frontal faces of mice, and estimate the pain level. We demonstrated that systems that have been successfully used for face analysis in humans can be adapted to animals. Our face detection model reached an IoU of 0.87. Our pain detection algorithm trained on these eye patches is also effective, providing promising results that are clearly better than baseline methods. We are able to discriminate between no pain (value 0) and pain (value 2) using predicted and actual bounding boxes for the mice's eyes. The proposed approach for this task achieves a performance of 97.2% in terms of accuracy. When we evaluated the performance with three classes, including the intermediate level pain (value 1), the performance dropped. A possible reason is the similarly between pain levels 1 and 2, which are difficult to distinguish between each other. Collectively, the evaluation shows promising results, which can be improved by collecting and annotating more frames.

For our future endeavors, we plan to extend this work to include other features such as ears, nose, whiskers and cheeks as additional attributes to better discriminate the pain levels. Some of these facial landmarks will require our system to focus on details in the image. A potential solution is to progressively resize the image to learn from coarse to detailed features (Bhatt et al., 2021). We can also explore deep learning methods that increase the interpretability of the model, indicating what facial cues were used to determine a given prediction.

Furthermore, we trained our models with limited annotated data. Using a semi-supervised model can be helpful in learning inherent structures in the data that can help us train more efficient models with the limited labeled data. An alternative approach is using more sophisticated data augmentation techniques (Chaudhari et al., 2020). Since we have continuous recordings of the mice in their cage, we want to explore longitudinal analysis correlating the predicted pain level with procedures conducted by the neuroscience team (e.g., measuring time needed for a drug to effectively reduce chronic pain). Finally, we would also like to work with mice with black fur (e.g., the C57BL6 strain), which is a more challenging problem, given that facial landmarks are harder to distinguish from their fur. For some problems, using handcrafted features can lead to better performance than models using feature representations directly learned from the data, especially for cases with limited data (Sanghani & Kotecha, 2019; Nanni et al., 2017). A future research direction is to explore if handcrafted features can be beneficial to detect facial landmarks of mice with black fur. This is an important goal because most mice used in pain research are on genetic backgrounds that have black fur (Mogil, 2009).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A system for largescale machine learning. In *Symposium on Operating Systems Design and Implementation (OSDI 2016)* (pp. 265–283). Savannah, GA, USA.

Baltrušaitis, T., Robinson, P., & Morency, L. P. (2013). Constrained local neural fields for robust facial landmark detection in the wild. In *IEEE International Conference on Computer Vision Workshops (ICCVW 2013)* (pp. 354–361). Sydney, Australia. doi:10.1109/ICCVW.2013.54.

Batista, J. C., Bellon, O. R., & Silva, L. (2019, October). YOLO-FD: YOLO for face detection. In Iberoamerican Congress on Pattern Recognition (pp. 209-218). Springer, Cham.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends* R *in Machine Learning*, *2*, 1–127. doi:10.1561/2200000006.

Bhatt, A., Ganatra, A. and Kotecha, K., 2021. COVID-19 pulmonary consolidations detection in chest X-ray using progressive resizing and transfer learning techniques. Heliyon, 7(6), p.e07211.

Burghardt, T., & Ćalić, J. (2006). Analysing animal behaviour in wildlife videos using face detection and tracking. *IEE Proceedings - Vision, Image and Signal Processing*, *153*, 305–312. doi:10.1049/ip-vis:20050052.

Chaudhari, P., Agrawal, H. and Kotecha, K., (2020). Data augmentation using MG-GAN for improved cancer classification on gene expression data. *Soft Computing,* 24(15), pp.11381-11391.

Chen, W., Huang, H., Peng, S., Zhou, C., & Zhang, C. (2021). YOLO-face: a real-time face detector. *The Visual Computer*, 37(4), 805-813.

Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. URL: https://keras. io/k, 7(8), T1.

Dalvi, C., Rathod, M., Patil, S., Gite, S., & Kotecha, K. (2021). A Survey of AI-Based Facial Emotion Recognition: Features, ML & DL Techniques, Age-Wise Datasets and Future Directions. *Ieee Access*, *9*, 165806-165840.

Descovich, K. A., Wathan, J., Leach, M. C., Buchanan-Smith, H. M., Flecknell, P., Framingham, D., & Vick, S. J. (2017). Facial expression: An under-utilised tool for the assessment of welfare in mammals. *Altex*.

Deuis, J., Dvorakova, L., & Vetter, I. (2017). Methods used to evaluate pain behaviors in rodents. *Frontiers in Molecular Neuroscience*, *10*, 1–17. doi:10. 3389/fnmol.2017.00284.

Garg, D. , P. Goel, S. Pandya, A. Ganatra and K. Kotecha, "A Deep Learning Approach for Face Detection using YOLO," 2018 *IEEE Punecon*, 2018, pp. 1-4,

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)* (pp. 580–587). Columbus, OH,USA.doi:10.1109/CVPR.2014.81.

Hassler, S., Ahmad, F., Burgos-Vega, C., Boitano, S., Vagner, J., Price, T., & Dussor, G. (2019). Protease activated receptor 2 (PAR2) activation causes migraine-like pain behaviors in mice. *Cephalalgia*, *39*, 111–122. doi:10.1177/ 0333102418779548.

Hewitt, C., & Mahmoud, M. (2019). Pose-informed face alignment for extreme head pose variation in animals. In *International Conference on Affective Computing and Intelligent Interaction (ACII 2019)* (pp. 1–6). Cambridge, UK. doi:10.1109/ACII.2019.8925472.

King, D. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations* (pp. 1–13). San Diego, CA, USA.

Kopaczka, M., Ernst, L., Heckelmann, J., Schorn, C., Tolba, R., & Merhof, D. (2018). Automatic key frame extraction from videos for efficient mouse pain scoring. In *International Conference on Signal Processing and Integrated Networks (SPIN 2018)* (pp. 248–252). Noida, India. doi:10.1109/SPIN.2018. 8474046.

Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Kamali, S., Malloci, M., Pont-Tuset, J., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., & Murphy, K. (2017). Openimages: A public dataset for large-scale multi-label and multi-class image classification. URL: https://github.com/ openimages.

Langford, D., Bailey, A., Chanda, M., Clarke, S., Drummond, T., Echols, S., Glick, S.,

Ingrao, J., Klassen-Ross, T., LaCroix-Fralish, M., Matsumiya, L., Sorge, R., Sotocinal, S., Tabaka, J., Wong, D., van den Maagdenberg, A. M. M., Ferrari, M., Craig, K., & Mogil, J. (2010). Coding of facial expressions of pain in the laboratory mouse. *Nature methods*, *7*, 447. doi:10.1038/nmeth. 1455.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. (2016).

SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *European Conference on Computer Vision (ECCV 2016)* (pp. 21–37). Amsterdam, the Netherlands: Springer Berlin Heidelberg volume 9905 of *Lecture Notes in Computer Science*. doi:10.1007/ 978-3-319-46448-0_2.

Lu, Y., Mahmoud, M., & Robinson, P. (2017). Estimating sheep pain level using facial action unit detection. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)* (pp. 394–399). Washington, DC, USA. doi:10.1109/FG.2017.56.

McLennan, K., & Mahmoud, M. (2019). Development of an automated pain facial expression detection system for sheep (ovis aries). *Animals*, *9*, 196. doi:10.3390/ani9040196.

McLennan, K., Rebelo, C., Corke, M., Holmes, M., Leach, M., & ConstantinoCasas, F. (2016). Development of a facial expression scale using footrot and mastitis as models of pain in sheep. *Applied Animal Behaviour Science*, *176*, 19–26. doi:10.1016/j.applanim.2016.01.007.

Mogil, J.S., (2009). Animal models of pain: progress and challenges. *Nature Reviews Neuroscience*, 10(4), pp.283-294.

Nanni, L., Ghidoni, S., & Brahnam, S. (2017). Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, *71*, 158-172.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)* (pp. 779–788). Las Vegas, NV, USA. doi:10.1109/CVPR.2016.91.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *ArXiv e-prints (arXiv:1804.02767)*, (pp. 1–5). arXiv:1804.02767.

Sanghani, G., & Kotecha, K. (2019). Incremental personalized E-mail spam filter using novel TFDCR feature selection with dynamic feature update. *Expert Systems with Applications*, *115*, 287-299.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)* (pp. 2818–2826). Las Vegas, NV, USA. doi:10.1109/CVPR.2016.308.

Tian, Y.-I., Kanade, T., & Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 97–115. doi:10.1109/34.908962.

Tuttle, A., Molinaro, M., Jethwa, J., Sotocinal, S., Prieto, J., Styner, M. A., Mogil, J., & Zylka, M. (2018). A deep neural network to assess spontaneous pain from mouse facial expressions. *Molecular Pain*, *14*, 1–9. doi:10.1177/ 1744806918763658.

Van Loon, J., & Van Dierendonck, M. (2018). Objective pain assessment in horses (2014-2018). *The Veterinary Journal*, *242*, 1–7. doi:10.1016/j.tvjl. 2018.10.001.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)* (pp. 511–518). Kauai, HI, USA volume 1. doi:10.1109/CVPR.2001.990517.

Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR 2016)* (pp. 1–13). San Juan, Puerto Rico.

Zeiler (2012). ADADELTA: An adaptive learning rate method. *ArXiv e-prints (arXiv:1212.5701)*, (pp. 1–6). arXiv:1212.5701.

Zhao, K., Chu, W. S., la Torre, F. D., Cohn, J. F., & Zhang, H. (2016). Joint patch and multi-label learning for facial action unit and holistic expression recognition. *IEEE Transactions on Image Processing*, *25*, 3931–3946. doi:10.1109/TIP.2016.2570550.

Zeng, H. (2022). Real-Time Traffic Sign Detection Based on Improved YOLO V3. In *Proceedings of the 11th International Conference on Computer Engineering and Networks* (pp. 167-172). Springer, Singapore.

Zeng, W., & Amen, B. (2021, December). Applications of Mobile Machine Learning for Detecting Bio-energy Crops Flowers. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 724-729). IEEE.