Automated Rat Grimace Scale for the Assessment of Pain

Brendan Elliot Arnold

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Biomedical Engineering

Pamela J. VandeVord, Chair

Sujith Vijayan

Chris L. Thomas

April 27, 2023

Blacksburg, Virginia

Keywords: Rat Grimace Scale, Pain, Behavior, Machine Learning

Automated Rat Grimace Scale for the Assessment of Pain

Brendan Elliot Arnold

Abstract

Pain is a complex neuro-psychosocial experience that is internal and private making it difficult to assess in both humans and animals. In research approximately 95% of animal models use rodents, with rats being among the most common for pain studies [3]. However, traditional assessments of the pain response struggle to demonstrate that the behaviors are a direct measurement of pain. The rat grimace scale (RGS) was developed based on facial action coding systems (FACS) which have known utility in non-verbal humans [6, 9]. The RGS measures facial action units of orbital tightening, ear changes, nose flattening, and whisker changes in an attempt to quantify the pain behaviors of the rat. These action units are measured on frontal images of rats with their face in clear view on a scale of 0-2, then summed together. The total score is then averaged to find a final value for RGS between 0-2. Currently, the software program Rodent Face Finder® can extract frontal face images. However, the RGS scores are still manually recorded which is a labor-intensive process, requiring hours of training. Furthermore, the scoring can be subjective, with differences existing between researchers and lab groups. The primary aim of this study is to develop an automated system that can detect action unit regions and generate a RGS score for each image. To accomplish this objective, a YOLOv5 object detector and Vision Transformers (ViT) for classification were trained on a dataset of frontal-facing images extracted using Rodent Face Finder®. Subsequently, the model was then validated using a RGS test for blast traumatic brain injury (bTBI). The validation dataset consisted of 40 control images of uninjured rats, 40 images from the bTBI study on the day of injury, and 40 images 1-month post-injury. All 120 images in the validation set were then manually graded for RGS and tested using the automated RGS system. The results indicated that the automated RGS system accurately and efficiently graded the images with minimal variation in results compared to human graders in just $1/14^{th}$ of the time. This system provides a fast and reliable method to extract meaningful information of rats' internal pain state. Furthermore, the study presents an avenue for future research into real-time pain monitoring.

Automated Rat Grimace Scale for the Assessment of Pain

Brendan Elliot Arnold

General Audience Abstract

Pain is a difficult experience to measure, both in humans and animals. It can be a subjective experience that is largely based on individual perception and interpretation. Furthermore, in animals, pain is even more challenging to assess because they cannot communicate their experience through language. Nonetheless, animal research plays an important role in understanding and treating the underlying mechanisms of pain. In animal research, rats are commonly used to study pain. However, traditional methods of assessing pain behaviors are not meant to observe the pain experience, but instead analyze a response to an external stimulus. The rat grimace scale (RGS) was developed as a direct measurement of the pain experience by analyzing the facial features. Currently, RGS scores are manually recorded by trained researchers which is time-consuming and can be subjective. This study aimed to develop an automated system to identify pain related facial expressions and generate a RGS score for frontal-images of rats. The system was trained using a dataset of frontal-facing rat images with varying levels of RGS scores and validated using images of rats from a traumatic brain injury study. The results showed that the automated RGS system accurately identified RGS pain level differences between recently injured rats, uninjured rats, and rats which were allowed to recover for 1-month. Furthermore, the system provided a fast and reliable method for measuring rat pain behavior when compared to manual grading. With this system, researchers will be able to efficiently perform RGS test. Additionally, this study presents an opportunity for future automation of other grimace scales as well as research into real-time pain monitoring.

Acknowledgements

I would like to take this opportunity to express my gratitude to everyone who has supported me throughout my thesis research and time in Blacksburg. This research was partially funded by VT Institute for Critical Technology and Science.

First and foremost, I would like to express my deepest gratitude to my committee chair and advisor, Dr. VandeVord. Thank you for allowing me to work in your research lab throughout my undergraduate years and into my master's research. Your guidance and support have been instrumental in shaping my future. I've been allowed to explore a wide array of research topics and find out what I enjoy. I am truly grateful for the opportunity you have provided me and hope to continue working with you in the future.

To my committee members, Dr. Vijayan and Dr. Thomas, thank you for taking the time to meet with me and provide valuable advice and support as committee members. To Dr. Thomas, thank you for helping me with your expertise in computer vision. Without your advice I would not have had the opportunity to make a truly meaningful contribution to pain research.

To my colleagues at the TNT lab, thank you for providing a wonderful work environment, and for all your support throughout the last year. I would like to thank Dr. Murphy, Amirah, and Caiti-Erin for helping with animal testing. Additionally, I would like to thank the master's corner for always putting up with my talking. I have enjoyed working on a projects togethers from undergraduate to graduate school. I wish you the best of luck wherever the next few years take you. I would like to especially thank my undergraduate assistant Rahul for bringing forward the idea of this project and helping execute it. I would also like to thank Kelsey for always providing technical and emotional support. Thank you for introducing me to the world of computer vision, without it this paper would not exist. Your encouragement and guidance have been invaluable in helping me become a better scientist and engineer.

To the friends that I made in Blacksburg over the last five years, I am truly grateful for the time we have spent together. There are too many of you to name individually, but please know I appreciate each and every one of you.

Lastly, I want to thank my family. Mom, Dad, and Josh I don’t know where I can start, but without you I would not be the person I am today. I cannot express how much I appreciate everything you have done for me throughout my life. Your love and support mean the world to me. I truly love you unconditionally.

To all of those who have supported me on this journey, thank you! Your support has made this research possible.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

bTBI - Blast Traumatic Brain Injury

CNN - Convolutional Neural Network

CSP - Cross Stage Partial Network

FACS - Faction Action Coding System

FN - False Negative

FP - False Positive

GELU - Gaussian Error Linear Units

GIoU - Generalized Intersection over Area

HOG - Histogram of Oriented Gradients

ICC - Intraclass Coefficient Correlation

KNN - K-Nearest Neighbors

MGS - Mouse Grimace Scale

MLP - Multi-Layer Perceptron

NLP - Natural Language Processing

NN - Neural Network

PANet - Path Aggregation Network

RGS - Rat Grimace Scale

ReLU - Rectified Linear Unit

R-CNN - Regions-based CNN

SSD - Single Shot Detector

SPP - Spatial Pyramid Pooling

SGD - Stochastic Gradient Descent

SVM - Support Vector Machines

TN - True Negative

TP - True Positive

ViT - Vision Transformer

YOLO - You Only Look Once

Introduction

Pain is a complex multifaceted experience influenced by a range of biological, psychological, and social factors. Furthermore, it is a subjective experience that can be challenging to quantify and measure objectively in both humans and animals [1,2]. Despite the difficulties associated with assessing pain, it is a crucial aspect of research elucidating the mechanisms of injuries and their potential treatments. This research can encompass studies on pain mechanisms such as, the role of neurotransmitters, nerve pathways, and other biological processes, as well as research on potential instruments to alleviate the underlying processes. However, conducting research on human subjects is very challenging. This is partly due to ethical considerations and limited ability to congregate a group of subjects with similar conditions without pronounced confounding variables. Nevertheless, advancing our understanding of pain and developing new treatments must be conducted to provide a better quality of life for millions worldwide. Therefore, animal research is paramount in investigating pain's mechanisms for improved treatment.

In preclinical research approximately 95% of animal models use rodents, with rats being among the most common for pain studies [3]. Rat models are commonly used in pain research due to the many similarities they share with humans in their physiology and pain pathways. These similarities enable researchers to study injury response and treatment in rats and to relate the finding to human response, prior to clinical studies. Despite the advantages of rodents in research, they are still non-verbal beings that cannot directly communicate their experiences. To compensate for this limitation, researchers use a range of tests to assess the effectiveness of a treatment or the extent of an injury. These include assessments of response to nociceptive stimuli such as mechanical, thermal, and chemical responses as well as behavioral tests including the tracking of movement, weight loss, and socialization [4]. While these tests can provide useful information on the wellbeing of the rat, traditional assessments of the pain response struggle to demonstrate a direct connection between behavior and pain experience. In many of these tests, the secondary response to pain is monitored such as latency for withdrawal from a stimulus, rather than the pain experience itself.

One solution for this problem is the use of Facial Action Coding Systems (FACS). FACS are coding systems which analyze facial body language such as eyebrow movement, orbital tightening, and other muscular movements, to determine emotional expression [5]. By assessing specific facial movements under known emotional contexts researchers can identify particular action units for any emotional expression and use them as a systematic measurement to determine emotional states without relying on verbal communication. FACS have demonstrated utility among nonverbal humans who may have cognitive impairments or other communication disabilities as a means of communication [6-8]. Building on the success in nonverbal humans, further coding systems were developed for a variety of animals, including laboratory rats, to provide a direct measurement of pain.

The rat grimace scale (RGS) was developed to directly quantify the rat's pain response using facial expressions [9]. RGS uses four action units including orbital tightening, nose flattening, ear changes, and whisker changes and scores them on a 0-2 scale to numerically evaluate the rat's wellbeing. During a RGS experiment, the rat is placed in a clear chamber, and they are recorded with a frontal-facing camera. The frontal images of the rat are then taken from specified timepoints throughout the video and individually graded by a team of trained researchers. This technique was originally developed for studying acute pain instances; however, it has been used for chronic pain studies as well [10, 11]. The RGS has been employed in a variety of research settings encompassing studies including, but not limited to, anesthesia methodologies, chronic inflammation, and brain injuries. Despite its broad applications, the use of RGS can be challenging due to its labor-intensive process and the requirement of a team of trained researchers. This limitation creates a barrier to many researchers who may want to use this technique but struggle to find a way to start. Additionally, while RGS grading between researchers has shown satisfactory inter-rater agreement, differences still occur due to the subjective nature of grading action units within gray areas [12]. To combat these obstacles a tool could be developed using computer vision techniques for automated RGS grading.

The application of computer vision and machine learning has demonstrated significant success in a large variety of behavioral tracking applications for humans and animals [13,14]. This includes programs which monitor FACS for humans, mice, and cats [13, 15, 16]. However, no fully automated system has been developed for RGS despite the widespread use of rats in pain

studies. This study aims to address this gap by developing an automated RGS scoring system based on object detection and classification models commonly used in computer vision. The first objective of this study was to create an automated RGS scoring system that can detect action units within an image and grade them according to the RGS standards. The second aim was to then validate this program against hand graded RGS images for accuracy and efficiency. The development of an automated system for RGS scoring would greatly enhance the efficiency of RGS as well as reduce the barriers to entry which may prevent studies from utilizing the RGS system.

Background

Grimace Scales

The history of using facial expression as a pain indicator dates as far back as the 19th century where Charles Darwin claimed that animal emotional states could be detected through facial expressions [17]. This idea was further developed with the advent of FACS, which had researchers identify and quantify specific action units in human facial expressions to predict a person's state of emotion. This practice was specifically extrapolated towards pain where the action units of lowering eyebrows, eye squeezing, nose wrinkling, opening of the mouth, and a raised upper lip were used to identify pain without verbal communication [18]. The application of FACS has also been expanded to include nonverbal infants, where the use of anesthesia and pain medication for neonates were justified due to pain related facial expressions [19]. Despite the major differences observed between animals, similar action units were observed for pain related expression in a variety of animals [20].

Langford *et al.* (2010), conducted one of the earliest studies to evaluate pain response in non-human mammals, using laboratory mice [21]. This study performed a preclinical pain assay of 0.9% acetic acid abdominal constriction test to determine pain related action units and develop a mouse grimace scale (MGS). After introducing the pain stimulus, the mice were filmed for 30 minutes, and facial images were captured at every 3-minute increment. Using these images, they created a FACS to measure pain expression through orbital tightening, nose bulging, cheek bulging, ear positioning, and whisker changes. A secondary study was then conducted where the MGS was applied to a mouse migraine model which found an elevated MGS score in the mutant mice compared to wild-type mice. The success of the MGS led to the development of grimace scales for other species, including felines, horses, and pigs [22-24]. However, the most significant development was the RGS, which provided researchers with a primary measure of pain in one of the most prevalent pain models in research [3, 9].

The RGS was developed shortly after MGS in a study by Sotocinal *et al.* (2011) [9]. Sotocinal and colleagues aimed to replicate the success seen in the MGS by using a FACS specifically modified for rats. The researchers found four action units that dictated a pain response through the use of an inflammatory assay and laparotomy study. This included orbital

tightening, ear changes, nose flattening, and whisker changes. Additionally, the researchers went on to partially automate the RGS process by automating the collection of frontal-facing images in a program called Rodent Face Finder®.

According to Mogil *et al.* (2020), the usage of the RGS and the MGS has been on the rise since their development [20]. Initially designed for acute pain, RGS has been applied to chronic pain studies, where it has shown elevated scores long after the initial stimulus [10, 11]. Despite its effectiveness, using RGS can be challenging due to the time-consuming image collection, grading, and training involved. Additionally, research has found that inter-rater reliability of RGS, measured by the intraclass correlation coefficient (ICC), is high for orbital tightening but less so for other action units [11, 12, 25]. Increasing the number of graders might address this issue, but the required training and time commitment makes this difficult to implement. Moreover, many studies using RGS do not provide details about the rater's training or inter-rater agreement, undermining the reliability of their results [20]. For these reasons research into machine learning and computer vision techniques for grimace scale grading is currently being explored.

Several grimace scales are exploring the automation of pain detection through facial landmarks [16]. However, the largest focus has been on the MGS, which has an automated version (aMGS) based on a deep convolutional neural network (CNN) [15]. Furthermore, additional research has taken place to improve upon the developed model [26, 27]. Despite the resources being implemented into the improvement of the MGS system, very little work has been done to automate the RGS. Additionally, the current image collection tool Rodent Face Finder®, relies on outdated object detection techniques such as Haar cascades. Therefore, developing a fully automated RGS system could provide substantial benefits for pain research.

Image Classification

The foundation of computer vision is to replicate the human capability of recognizing visual information. At the forefront of this research is the classification of images into categories based on their visual content. Many machine learning techniques have been used to classify images into their respective categories, however, at the simplest level, most image classification

follows some form of feature extraction which is then run through a classification algorithm for pattern recognition.

In computer vision, many models are used for image classification including support vector machines (SVM), K-Nearest Neighbors (KNN), Naive Bayes, and deep learning to name a few. Each model uses a different algorithm to make predictions. In SVMs, a model finds hyperplanes that can separate the predicted algorithm output into categories, whereas KNNs find the K number of images with the closest resemblance in features and predicts the input based on the nearest neighbors in the training data [28, 29]. Additionally, some models use probability like Naive Bayes classifiers, which uses the Bayes' theorem to estimate the probability of a class given the features in an image [30]. Despite all of the potential methods for image classification, the development of neural networks (NN) and deep learning led to massive improvements in image classification for complex models [31].

Neural networks, illustrated in Figure 1, are inspired by the structure and function of the human brain [32]. A NN operates in a comparable way to a neuron by utilizing interconnected nodes to analyze data and make decisions. In general, these models use an input layer that receives raw data, which is then processed through the hidden layers and passed towards the output layer to make a prediction. Each node in the network receives inputs from the previous layer and applies an activation function, which introduces nonlinearity within the model. This activation function is used to discover complex features in the input. The output of each node is then multiplied by a set of weights and passed on to the next layer of nodes. The product of the activation function's output and the weights are calculated for each node in the next layer, and the process is repeated until the final layer which produces the network's output.

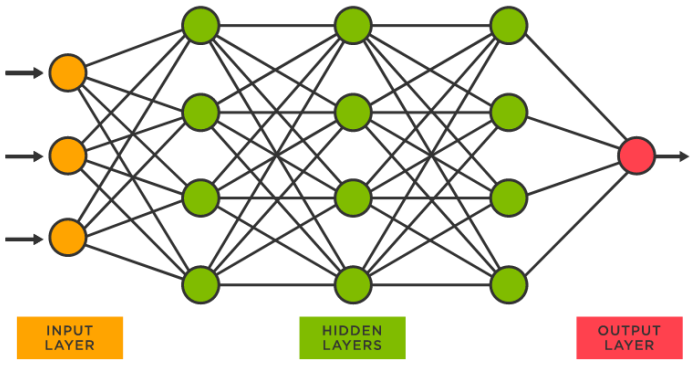

Figure 1: *A basic neural network architecture courtesy of TIBCO [33].*

One advantage of a NN is its training process, in which the weighted sums are optimized to recognize patterns and determine the importance of each node's output for the next layer. These weights are optimized through training data, which in the case of supervised learning has a known label. As the network predicts the training data, the optimization algorithm adjusts the weights such that the prediction accuracy on training data increases. This process results in a final output layer which makes informed decisions based on which nodes are firing due to the input it receives.

Convolutional neural networks are commonly used for image recognition due to their effectiveness in analyzing spatial features in grid-structured data [34]. The general architecture of CNNs, visualized in Figure 2, consists of convolutional and pooling layers for feature extraction and fully connected layers for classification. Convolutional layers are responsible for developing feature maps from an image while the pooling layers are used to down-sample the feature maps. Convolutional layers work by using a set of filters, also known as kernels, to scan over an image and produce a feature map. Each filter can capture different local patterns within an image such as vertical or horizontal edges. If a feature is detected within a local region the kernel will produce an increased output to the next layer similarly to normal neural networks. This process is repeated for each local region of an image until a feature map is generated. The data is then run through a pooling layer which down-samples the feature maps, in the case of max pooling the layer takes the highest weight for each area on an image. This technique saves the feature map weights while reducing the spatial dimensions which saves computational time. This process can then be repeated multiple times within a CNN to extract complex features from an image.

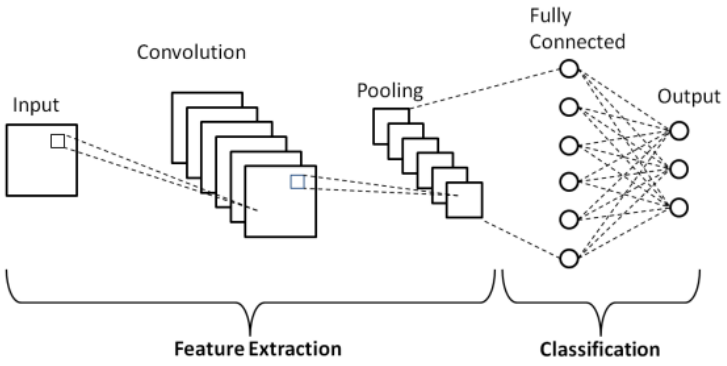

Figure 2: *A Simple convolutional neural network architecture courtesy of Phung et al. [35].*

Once the internal feature extraction is performed, the final pooling layer is flattened into a one-dimensional array before it is fed into to the fully connected layers. These layers act similarly to the hidden layers in a traditional neural network and result in an output layer which determines the image's classification. One key benefit of CNNs is their flexibility, which allows changes to the internal architecture to try and enhance the networks' performance for its task [32]. This has led to the development of many image classification competitions, which commonly drive innovation in CNN architecture. Furthermore, these competitions also result in the creation of large open-source datasets that can be used to pre-train models prior to fine-tuning for specific tasks.

A recent advancement in computer vision is from the transformer architecture, which was originally developed for natural language processing (NLP). The transformer architecture was developed in a landmark paper by Vaswani *et al.* (2017) and has revolutionized the NLP field for its performance due to its use of the attention mechanisms [36]. In NLP, the transformer receives input data as a sequence of words, known as tokens, each with a sequential value for its position in the input. The model then uses self-attention layers to compute the relationships between different parts of the sequence and assigns attention values to each token, which determines its importance in making decisions. Similarly, to CNNs, this internal architecture is flexible and allows for multiple attention layers to further evaluate complex relationships between the tokens. After the attention layers, the transformer model uses feed-forward layers to process the attention values. The feed-forward layer performs a linear transformation of the data to extract different information such as local and global relationships within the tokens.

The vision transformer (ViT) architecture, pictured in Figure 3, was introduced by Dosovitskiy *et al.* (2020), and is a recent development in computer vision that replaces the traditional CNNs with the transformer architecture that was originally developed for NLP [37]. This model follows the same procedure as transformers in NLP, where it uses tokens and compares the positional relevance to make decisions. However, instead of using a sequence of words as tokens, the ViT uses patches of the image as the input tokens. The original ViT design uses patch sizes of 16 x 16 pixels, so if an input image is 224 x 224 pixels, a sequence of 196 tokens is used within the transformer. Each token is then flattened into a linear sequence of pixels and run through the transformer encoder which usually consists of self-attention layers

and feed-forward layers called multi-layer perceptron (MLP). These layers act as a replacement for the convolution layers in a CNN and extract features from the input tokens. The final MLP acts as a dense layer before producing a prediction score for each classification output. One advantage of ViTs over traditional CNNs is their ability to process all of the data in parallel, which makes them computationally faster and usable for real-time image classification.

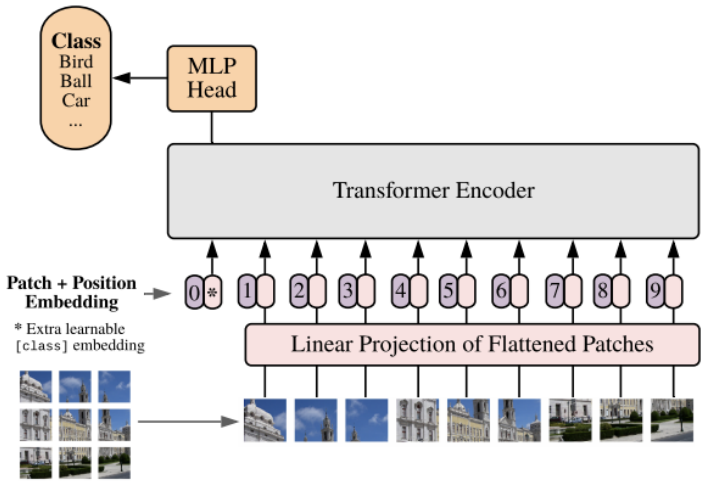

Figure 3: *Vision transformer model overview courtesy of Dosovitskiy et al. [37].*

Object Detection

In computer vision, object detection extends beyond a simple classification problem, by not only identifying an object, but also determining its location within an image. Before the emergence of deep learning techniques, object detection was commonly achieved through a feature-based approach that relied on identifying specific characteristics within an image to detect the presence of an object. One of the earliest and most notable implementations of object detection was created in Viola & Jones (2001), which introduced the Haar cascade algorithm [38]. This method uses Haar features which are simple rectangular filters that scan an image searching for features such as an edge, corner, or lines. Originally, Haar features were manually selected, however, Haar cascades can now be trained to generate their own features using positive and negative samples of the object. The Haar cascade then works by applying a sequence of Haar filters, which can either approve or reject an image based on the presence of specific features. This process can be repeated using a sliding window across an image to find

the location of an object. Despite its innovation for its time, Haar cascades have many limitations that hinder its use in modern computer vision. These include the tendency to produce false negatives and sensitivity to image scale, brightness, and object orientation.

Another early object detector called the histogram of oriented gradients (HOG) was introduced in Dalal and Triggs (2005) [39]. HOG detectors are based on pixel gradient orientation and change in intensity within neighboring pixels. This technique divides an image into smaller cells and calculates the gradient of pixel change within each cell. The values produced for intensity of change and gradient orientation can then be used as features for object detection. Similarly, to Haar cascades, a sliding window can be utilized on the HOG image to predict where an object is located within an image. Each image of the sliding window is then evaluated using a machine learning algorithm to predict whether an object is in the window. In the original paper Dalal and Triggs used an SVM, however, HOGs can also be used in various other models including neural networks [40]. While HOGs have improved on capturing object shapes and handling different object scales and orientations, they are computationally slow, making real-time applications challenging. Nonetheless, advancements in deep learning have led to improvement in both accuracy and processing time for object detection, overcoming the limitations of both Haar cascades and HOGs.

After the development of Haar cascades and HOGs, object detection algorithms had limited improvements until the advent of deep learning for image analysis. One of the earliest attempts to use CNN for object detection was proposed in Sermanet *et al.* (2013), known as OverFeat [41]. This method utilized a sliding window approach to propose object location. However, the approach was computationally expensive since it required running CNN over multiple locations and scales.

It was not until the development of the object detector, Region-based CNN (R-CNN) from Girshick *et al.* (2014) and the subsequent Faster R-CNN from Ren *et al.* (2015), that deep learning marked a substantial improvement over earlier object detection methods [42, 43]. The original R-CNN works by initially proposing specific regions within an image, then running the proposed regions through a CNN for feature detection and finally classifying the region using an SVM. In the initial paper Girshick *et al.* (2014), region proposal was done using a selective search to partition an image into different region based on image structure [42]. This

methodology allowed for more efficient object detection since image classification was only performed for the proposed regions rather than computing the image classifiers over a sliding window. Soon after the release of the R-CNN, the same authors addressed some of the limitations by developing the Fast R-CNN in Girshick *et al.* (2015) [44]. This model worked to improve the efficiency of the initial R-CNN model by computing a single convolutional feature map on the entire image, then using proposed regions to predict an object through fully connected layers. While the Fast R-CNN marked better results than the initial R-CNN, it was quickly overtaken by the Faster R-CNN [43]. Faster R-CNN improved on its predecessors by implementing a region proposal network (RPN), rather than using an external method of region proposal such as selective search. The RPN works by taking the feature map and proposing anchor boxes which are then adjusted to fit the object. The proposals are then filtered to only perform classification on the highest scoring anchor boxes which significantly reduces region proposals. This method helped progress object detection closer to real-time implementation.

Despite R-CNN's impact on object detection, the emergence of more state-of-the-art methods improved upon accuracy and efficiency for object detection. One such model is the You Only Look Once (YOLO) architecture, which eliminated the need for region proposals and notably improved efficiency by predicting bounding boxes with a single forward pass through the neural network. The first YOLO model, YOLOv1, was introduced in Redmon *et al.* (2015) with the aim of unifying all parts of an object detection model into a single neural network [45]. The YOLO models work by dividing the image into a specified grid size, where each grid cell predicts a set of bounding boxes and corresponding confidence scores. This approach has shown to be highly effective, with YOLO models outperforming traditional object detection methods in both speed and accuracy.

In order to unify object detection into a single model, YOLO treats the problem as a regression rather than a classification output like its predecessors. The output of YOLOv1 is encoded as a tensor with $S\ x\ S\ x\ (B * 5\ +\ C)$ size, where $S\ x\ S$ is the grid size, B is the number of bounding boxes per grid cell, and C is the number of classes. The last portion of the tensor represents the number of bounding boxes times the 5 outputs of the (x, y) center coordinates, the width, the height, and the confidence score plus C which contains the values of each class probability. By running the image through the network once for all boxes and classes, the YOLO

architecture created one of the first accurate methods for real-time deep learning object detection. Additionally, the YOLO model architecture has been continuously improved since its initial creation, for efficiency, box precision, and accuracy. The same group created YOLOv1 through YOLOv4, however, other groups have begun releasing spinoff models that take inspiration from the original YOLO architecture [46-50]. These models have each demonstrated improvements upon the initial YOLO series and are still being released, with YOLOv8 by Ultralytics, as well as other spin off models all being released within the last year [50].

While other modern object detection models such as single shot detection (SSD), ViT object detectors, and newer spinoffs of YOLO have been developed, YOLOv5 was chosen for this project due to its ease of use, speed, and accuracy [46]. YOLOv5 by Ultralytics is a spinoff model based on the architecture of the original YOLO series and is illustrated in Figure 4. The YOLO architecture consists of three sections: the backbone, the neck, and the head.

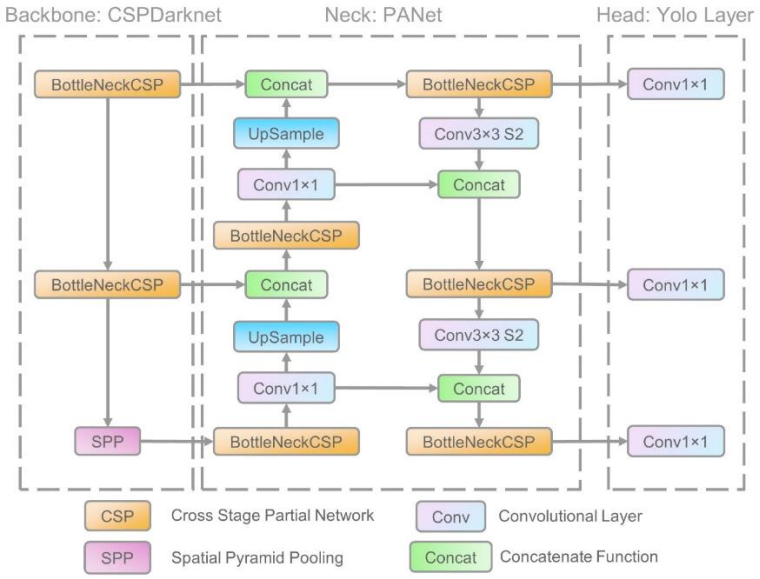

Figure 4: *YOLOv5 architecture courtesy of Xu et al. [51].*

YOLOv5 backbone uses CSPDarknet, which incorporates a cross stage partial network (CSP) into the original backend of the previous YOLO models, Darknet, and performs spatial pyramid pooling (SPP) before the data enters the neck [52-54]. The backbone starts by using CSP to split an image into two parallel pathways for the neck [52]. The original image is run through convolutional layers from Darknet to create a feature map and one of the pathways is held for fusion in the neck [53]. The second pathway is run through additional convolutional

layers and is once again split into two additional pathways where one is also held for fusion in the neck. The second pathway is then run through SPP which allows an image of any size to be used in the model [54]. By using a SPP layer, the model is able to detect objects at variety of scales without needing to perform more calculations by using multiple anchor box sizes.

The second section of the YOLOv5 model, the neck, fuses the various feature maps developed in the backbone of the model to use the data in the head of the model. The neck works by using a path aggregation network (PANet) which has additional CSP layers, as well as upsampling and concatenate functions to locate small objects and combine the feature maps [55]. The CSP layers within the neck are used to help capture both high-level and low-level features of the input image by splitting the image into two pathways which have different levels of feature maps. One of the pathways is then upsampled to increase the resolution of the feature map in order to detect smaller objects that may not have been originally seen in the model. After the new feature map is developed, it is then combined with the feature map from a previous section of the model using a concatenate function. This process allows the model to capture features at different scales to improve the overall detection ability of the model. Once this process is completed the final output of the neck is fed into three sections of the head using additional CSP layers.

The final stage of the YOLOv5 model, the head, takes in three different sizes of feature maps from the neck to predict objects at multiple scales. The shape of the head can change depending on the task the YOLOv5 model performs, however, it typically consists of a set of fully connected layers. After the feature maps run through the head of the model, a final output is produced in a similar manner to the original YOLO model, where it outputs a tensor containing class probabilities and bounding box coordinates for each detected object.

Methods

Database Collection

Frontal-facing images of white-furred rats were collected, and a database was constructed to develop a machine learning model for predicting RGS scores. The database included a total of 1,122 frontal-facing images. These images were used to further develop secondary databases containing 1,482 eye images, 1,363 ear images, and 1,117 nose images. While the original RGS included whiskers as an action unit, they had been found to be the least reliable for pain prediction [12, 56, 57]. Additionally, the variability of image quality in the dataset made the whiskers difficult to discern. For these reasons, a whisker grimace score was not included. The frontal images were collected using the Rodent Face Finder from Sotocinal *et al.* (2011), however, additional frames were extracted for instances of pain to help increase the number of examples for elevated RGS scores [9].

The frontal images were sourced from various RGS setups to ensure the prediction model could be generalized for different RGS imaging protocols. The majority of the database images include frontal images from Furman *et al.* (2020) where rats were induced with neuropathic orofacial pain by chronic constriction injury of the infraorbital nerve and Pang (2018) which developed an RGS dataset to evaluate interrater reliability. Additionally, the database was supplemented with frontal-facing rat images taken during Von Frey testing to increase the samples of elevated RGS action units. Von Frey tests are performed as an assessment of mechanical allodynia in animal models and elicit a pain response. In this study, the up-down method described in Dixon (1980), was used on the periorbital section of the head to determine the mechanical force required to elicit withdraw [60]. The Von Frey test was filmed using a GoPro HERO8 Black.

Image annotations were performed using Microsoft Excel for RGS scores. Two RGS graders were trained using the Pang Lab RGS training manual and the RGS scores for orbital tightening, ear changes, and nose flattening from Furman *et al.* (2020) and Pang (2018) [58, 59]. Once trained, the graders then scored the individual action units for RGS in the facial Von Frey testing. Shortly after, the images were then annotated for object detection using the open-source annotator Labelimg [61]. Bounding boxes were drawn around clearly visible ears and eyes that

could be viably used for RGS scoring. For the nose, a bounding box was drawn from the upper cheek bone below the eyes to the tip of the nose, as long as it was not obstructed.

Feature Detection

Action unit feature detection was performed by fine-tuning a YOLOv5 small (YOLOv5s) model. YOLOv5s was chosen due to its inference time of 98 ms per image on a CPU, as well as the 7.2 million parameters compared to 21.2 million parameters in the standard YOLOv5 model. The reduction in parameters was considered to prevent overfitting due to the limited dataset size. The model used a Leaky ReLU activation function $\varphi(x)$ within the hidden layers, and a sigmoid activation function $\varphi(x)$ for the final detection layer as illustrated in Eq. (1) and Eq. (2). In these equations, x represented the input features for the model layer and α (0.1) represented the hyperparameter for slope while x is negative [32, 62]. Additionally, the model used stochastic gradient descent (SGD) with momentum for the optimization function as depicted in Eq. (3). In this equation θ represented the models' parameters, which were updated by the momentum vector v_t [32]. The momentum vector was then updated each iteration from the previous vector $v_{t-1,}$ the momentum coefficient γ, the learning rate η (0.01), and the gradient of the loss function $\nabla_\theta J(\theta)$ from the current model parameters. The YOLOv5 model used warmup epochs where the momentum coefficient started at 0.8 and began to progress to 0.937 after the third epoch.

$$\varphi(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{1}$$

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$\begin{aligned} v_t &= \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \\ \theta &= \theta - v_t \end{aligned} \tag{3}$$

YOLOv5 calculated three loss functions including 1) box regression loss to determine how well the predicted bounding box covers the ground truth box 2) objectness loss to measure the probability an object is within the proposed region and 3) class loss for how well the algorithm is predicting the correct object class [63]. The objectness and class loss both used a binary cross-entropy loss function displayed in Eq. (4), where if no ground truth exists for a

predicted bounding box only objectness loss was affected. The loss was derived from the predicted probability $\hat{y}_i$, and the true label y_i, and the number of training samples *N*. The box regression loss was calculated using generalized union of intersection (GIoU), displayed in Eq. (5), where $\frac{|A \cap B|}{|A \cup B|}$ represented the intersection over union (IoU) which is a measure of the overlap between two bounding boxes *A* and *B* [64]. The second term $\frac{|C/(A \cup B)|}{|C|}$ represented the difference between the area of the smallest bounding box *C* that completely enclosed *A* and B when compared to the union of *A* and *B*. The overall loss function used in YOLOv5 was the weighted sum between the three losses, where box loss was weighted as 0.05, object loss was weighted as 1, and class loss was weighted as 0.5.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i) \tag{4}$$

$$Loss = \frac{|A \cap B|}{|A \cup B|} - \frac{|C/(A \cup B)|}{|C|} \tag{5}$$

The YOLOv5s model was pre-trained using the COCO database for 300 epochs. The COCO dataset contained 123, 287 images with 80 classes including various animals [65]. By pre-training the model, the parameter within YOLOv5s learned specific features from the COCO dataset and transferred that information to the action unit detection task.

After the model was pre-trained, it was fine-tuned for action unit detection using the collected images and their respective annotations. The model was trained for 100 epochs using mosaic augmentation shown in Figure 5. Mosaic augmentation combines four images into a single image and will randomly crop, resize, and change the brightness of the image [63]. The resulting mosaic provides a sample with a diverse set of objects within the image that vary in size and location. This method helps improve the robustness of the model to be able to detect auction units from various types of images. Additionally, this method helps prevent overfitting by increasing the diversity of the training data.

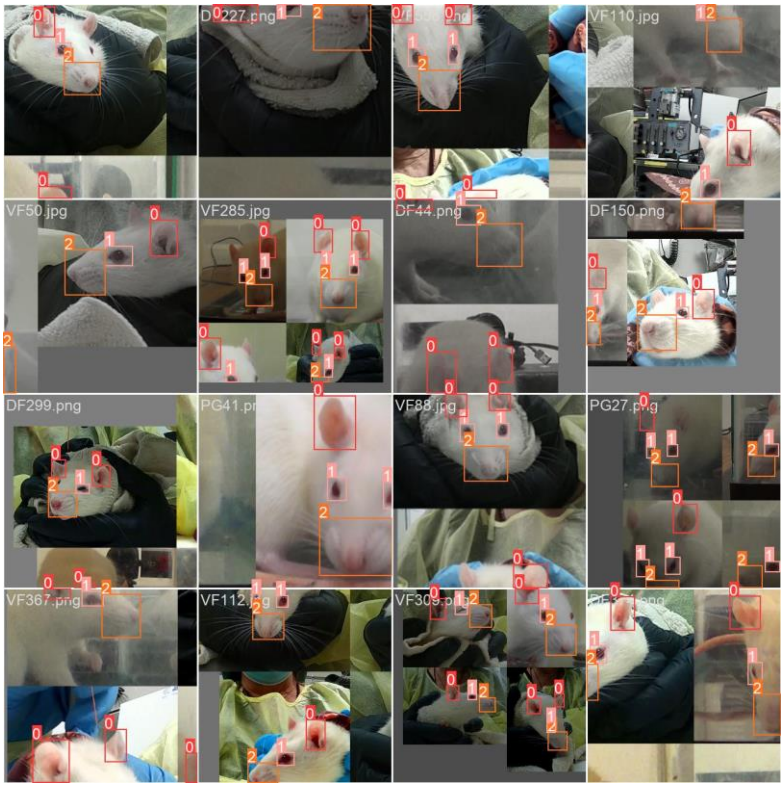

Figure 5: *Mosaic augmentation on the RGS dataset.*

Pain Classification

Individual classifier models were developed for action unit specific pain grading, where each action unit model was built to predict pain levels on a standard RGS scale of 0-2. The action unit classifiers were constructed using an open-source transformers package from Hugging Face [66]. The open-source ViT model utilized the architecture discussed in Dosovitskiy *et al.* and seen in Figure 3 [37]. In these models, the images are preprocessed to a size of 224 x 224 pixels and normalized. The ViT models employed a 16 x 16 patch size for a total of 196 patches and runs through 12 transformer encoders composed of self-attention and feed forward layers. The ViT uses SGD optimization, similar to the YOLO model, but with the addition of weighted cross-entropy loss due to a large class imbalance between the action unit RGS scores. Weighted cross-entropy follows the same function demonstrated in Eq. (4); however, it is multiplied to weights inversely proportional to the number of samples in each class to incentivize the model to learn the imbalanced classes equally. Additionally, the ViT uses

gaussian error linear units (GELU) for the activation function $\varphi(x)$, displayed in Eq. (6), where x represents the input features of the model layer [67].

$$\varphi(x) \approx 0.5x\left(1 + \tanh\left[\sqrt{2/\pi} \times (x + 0.044715x^3)\right]\right) \quad (6)$$

The ViT models were first pre-trained using the ImageNet-21k database, which contained 14 million images and 21,843 classes [68]. The models were then subjected to a fine-tuning process using the action unit databases to either the eye, ear, or nose, for a duration of 20 epochs. For training, 80% of each database was utilized, while the remaining 20% was reserved for validation. After training, the model state with the best validation loss was selected as the final action unit classifier within the automated RGS system. Moreover, a ViT model was trained for the complete frontal-facing image using a binary pain vs. no pain scale where RGS below =< 0.33 was no pain and => 1 was classified as pain. Additionally, ViTs were trained as binary classification for the 0 and 2 action units for further performance analysis.

Model Development

The holistic RGS model was developed to identify action unit areas within an image, provide individual action unit RGS grades, and output a total RGS score as demonstrated in Figure 6. In order to accomplish this goal, the program received a file path and coded all images within the file for RGS, ultimately producing an excel file that contained the filename, action unit grades, and the final RGS score. The program began by loading the object detection model and setting a minimum confidence criterion of 0.25 for detected objects, before proceeding to load each action unit specific classification model.

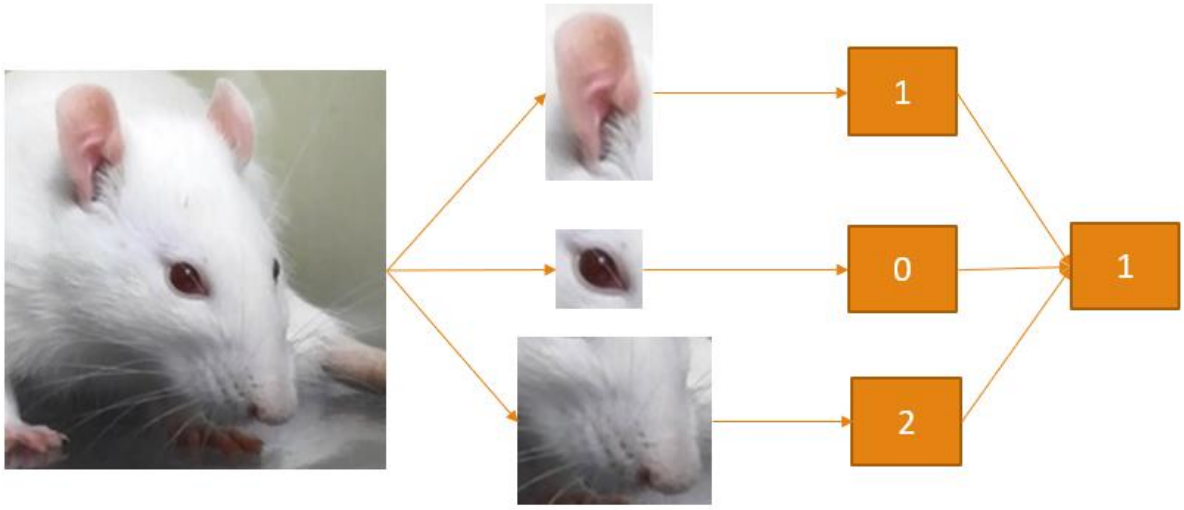

Figure 6: *RGS system pipeline [59].*

Upon receiving the file path, the program processed any .jpg or .png files through the RGS system pipeline, as displayed in Figure 6. In each image, the YOLOv5 model detected the action unit areas and outputs bounding boxes, class IDs, and confidence scores for each eye, ear, and nose. Subsequently, the program utilized the confidence score to take the two highest rated eyes and ears, and the highest rated nose in order to prevent false positive detections from adversely affecting the RGS score. The remaining detected objects were then sorted by class ID and cropped from the original image to prevent background noise from influencing the ViT models. The features were then processed in their respective ViT model classified as either 0 for no action unit, 1 for moderate appearance of an action unit, or 2 for an obvious appearance of an action unit [9]. Prior to computing a final RGS score, the program averaged the ears and eyes action unit groups if multiple objects were detected. The RGS score was finally recorded as the average of each action unit group, and if any action unit group found no detections from the YOLOv5 model, the action unit specific score and total RGS score is reported as NaN (Not a Number) in the excel sheet.

Model Analysis

An additional validation set was established using a previously validated injury model of blast traumatic brain injury (bTBI) with RGS testing [57, 69]. As part of the validation study, 10-week-old male Sprague Dawley rats were subjected to three static overpressure insults of 19 psi, followed by RGS imaging described in Sotocinal *et al.* [9]. The validation set was comprised of

40 control images of uninjured rats, 40 images from the existing bTBI injury study at the day of injury, and 40 images of 1-month post-injury taken with a GoPro HERO8 Black. This study was designed to ensure an adequate representation of injured and uninjured rats, as well as a longitudinal evaluation of the automated RGS model's efficacy in detecting pain behavior reduction over time.

The images were individually placed on separate PowerPoint slides and randomized by a third-party blinded to the original test groups. The images were then graded according to the RGS by the two trained graders for orbital tightening, ear changes, and nose flattening. After manually grading the images, the 120 samples were run through the automated RGS system and recorded for total computation time, total RGS score, as well as the granulated action unit scores.

Statistical analysis was conducted using GraphPad Prism 9 software. Due to the non-normal distribution of RGS data, the test groups were analyzed non-parametrically. A Wilcoxon test was used to compare the experimentally graded images to the model images. Furthermore, a Kruskal-Wallis test was used to compare the control images, to the 1-day and 1-month post-injury images, while an additional Wilcoxon test was used to compare the 1-day and 1-month groups. Differences were considered statistically significant if the P-value < 0.05.

Moreover, the intraclass correlation coefficients (ICC) were calculated between the model and the experimental validation grades, as well as the two graders using a one-way random effect, absolute agreement model depicted in Eq. (7) [70, 71]. In this equation MS_R is the mean square for rows, while MS_W is the mean square for residual sources of variance. The interpretation of the ICC was based on Landis & Koch (1977), where a score >0.81 was considered "very good", 0.8-0.61 was "good", 0.6-0.41 was "moderate", 0.4-0.21 was "fair", and anything <0.2 was "poor" [72].

$$ICC = \frac{MS_R - MS_W}{MS_R} \tag{7}$$

Additionally, model specific analysis was performed using python version 3.10.7 and MATLAB version R2021a. YOLOv5 and the ViT models were analyzed for precision and recall illustrated in Eq. (8) and Eq. (9), where the metrics are calculated using the number of true

positives (*TP*), false positives (*FP*), and false negatives (*FN*). Each ViT model was analyzed using scikit-learn confusion matrices and their respective weighted accuracy, shown in Eq. (10). In addition to the *TP*, *FP*, and *FN*, weighted accuracy also considers the true negatives (*TN*) as well as the number of samples in each class set N_n. Furthermore, the ViT models were visually investigated using attention heatmaps.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Weighted\ Accuracy = \frac{N_1 \left[\frac{TP + TN}{TP + TN + FN + FP}\right]_1 + \ldots N_n \left[\frac{TP + TN}{TP + TN + FN + FP}\right]_n}{N_1 + \ldots N_n} \tag{10}$$

Results

YOLOv5

The YOLOv5 model for action unit detection was effective in identifying the location and class of each action unit area in an image. The model was trained using the frontal-facing image database developed above to closely resemble real-world RGS images, as well as provide an array of facial orientations, image quality, and RGS scores. Additionally, the bTBI validation set was assembled using various testing setups to verify the model's generalizability. The model's performance was evaluated using this validation set, as shown in Figure 7.

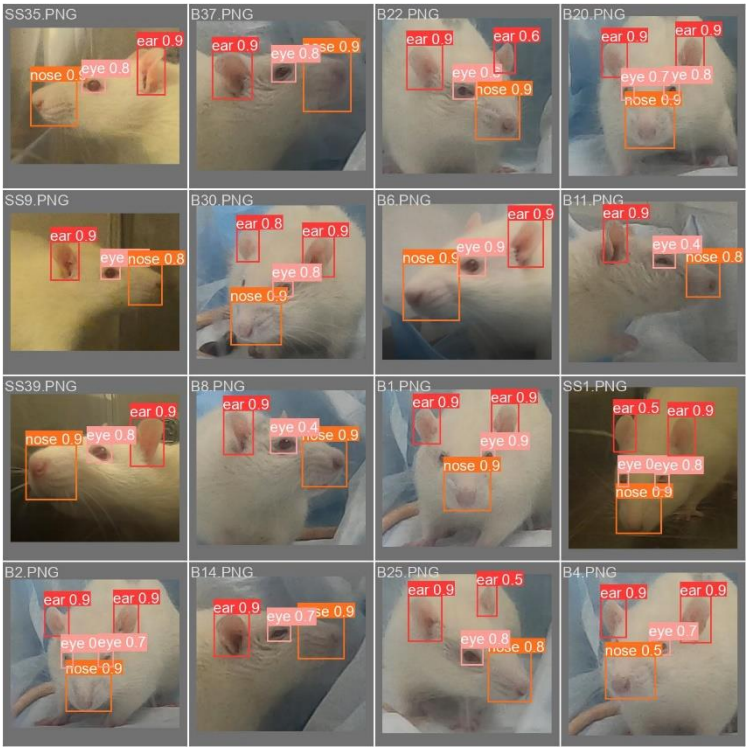

Figure 7: *YOLOv5 model predictions on a single batch of the bTBI validation set.*

The YOLOv5 model's progress was monitored by evaluating its three loss equations over 100 training epochs. Figure 8 displays the training and validation loss curves for box regression, objectness, and class losses, while Figure 9 demonstrates the holistic model loss value per epoch. The training loss curves are shown in blue, while the validation loss curves are shown in orange. The training loss steadily decreased for all three loss functions throughout the training process, indicating that the model was improving over time. However, the validation loss curves

generally stopped decreasing earlier in the training process. Nonetheless, the training loss remained greater than the validation loss until around the 70th epoch. To combat overfitting, the 68th epoch, which had the lowest total validation loss was taken as the final model for the action unit object detector.

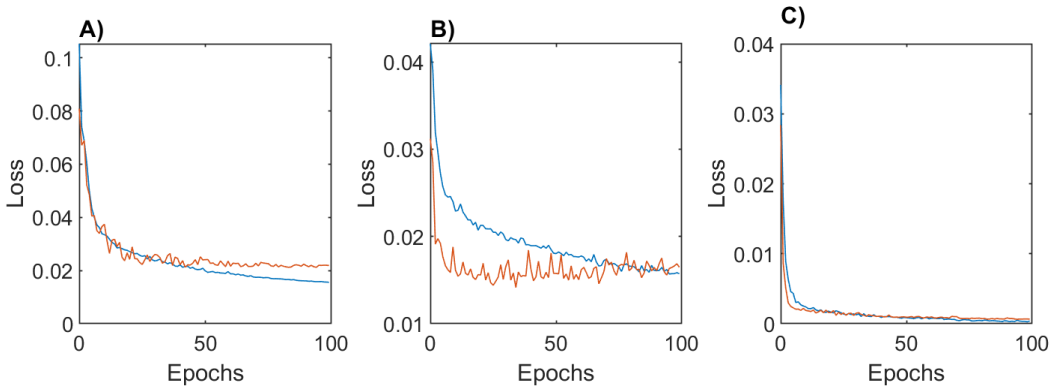

Figure 8: *YOLOv5 loss functions for the training set (blue) and validation set (orange) over the training epoch: A) Box regression loss; B) Objectness loss; C) Class loss.*

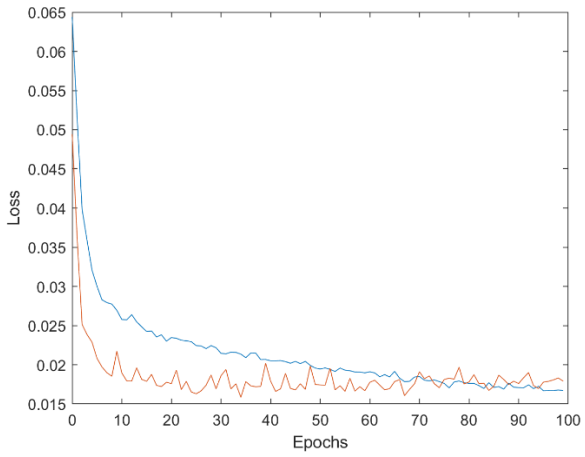

Figure 9*: Total YOLOv5 loss value for the training set (blue) and the validation set (orange) over the training epoch.*

The precision and recall metrics were analyzed over the course of the 100 training epochs to gain a better understanding of the model's performance, illustrated in Figure 10. Precision measured the proportion of true positives to the total positives detected, while recall measured how many true positives are found out of all ground truths. The results showed that both precision and recall quickly increased to 0.97 within the first 10 epochs and remained relatively stable throughout the rest of the training process. This indicated that the model was able to learn the training data and generalize the results to the validation set. The consistent scores suggested

that the model did not overfit during the training data and maintained its generalizability throughout the training process.

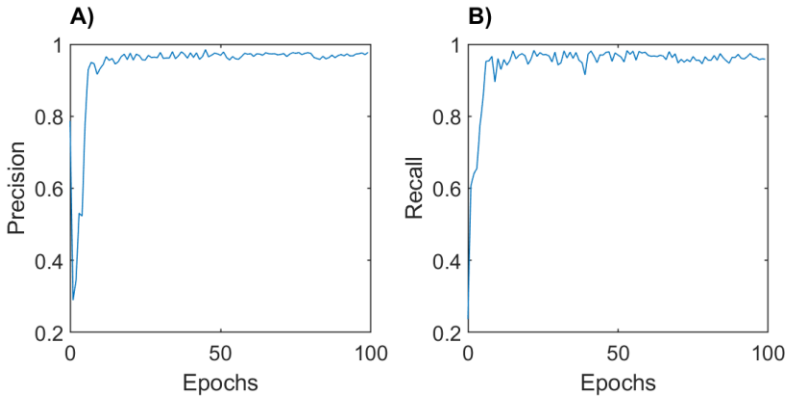

Figure 10: *Change in YOLOv5 metrics over training epochs: A) Precision; B) Recall.*

The precision and recall of the final model were analyzed in relation to confidence thresholds to evaluate its performance. Figure 11A displays the precision-confidence curve, which demonstrated how the precision increases as the model becomes more confident for its prediction. Additionally, Figure 11B shows the recall-confidence curve, illustrating that increasing the confidence threshold reduces the likelihood of detecting all objects in an image. These curves provided insight into the model's performance and helped determine the optimal confidence threshold. The precision-confidence curve displayed that the precision levels off around a confidence of 0.1 and showed minimal improvement until a confidence greater than 0.6. In contrast, the recall-confidence suggested that recall slowly decays as confidence increases until a confidence of 0.5, after which it rapidly drops off. Consequently, a confidence threshold of 0.25 was chosen to maintain a high level of recall without sacrificing significant precision. Recall was considered more important due to the ease of eliminating false positives within the

holistic RGS coding system, compared to having false negatives which leads to no RGS features for the downstream models to grade.

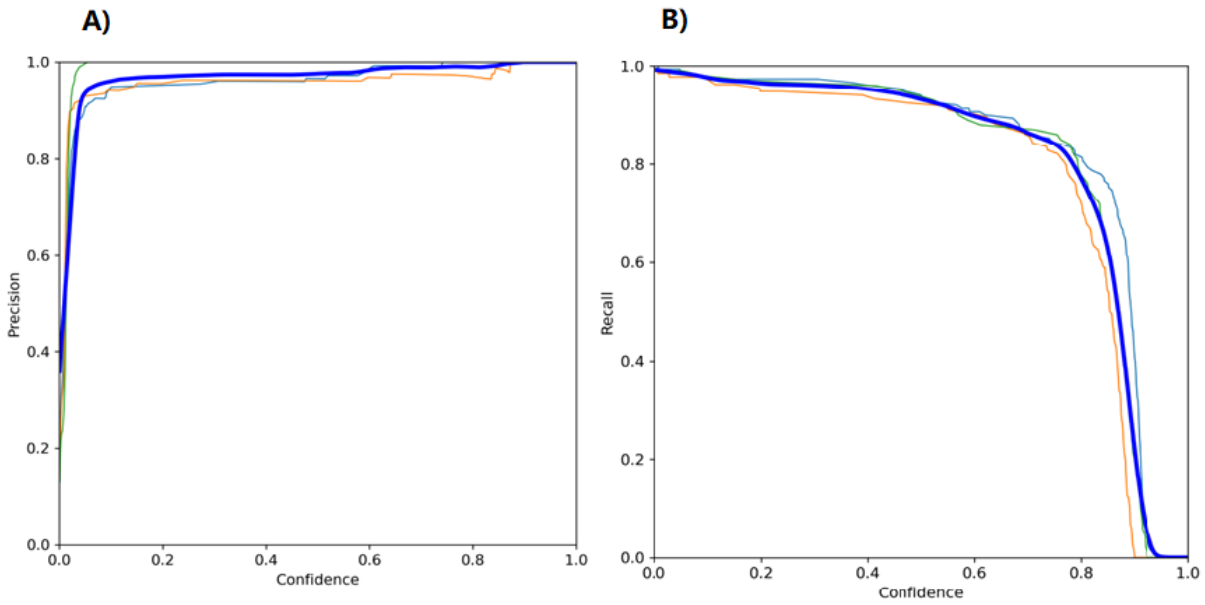

Figure 11: *YOLOv5 final model metrics for global features (bolded blue), eyes (orange), ears (light blue), nose (green): A) Precision-Confidence Curve; B) Recall-Confidence Curve.*

Vision Transformers

The ViT classification models displayed varying amounts of success in grading RGS pain scores. Figure 12 depicts the confusion matrices for each action unit classifier on the validation sets. The models performed well in identifying differences in the appearance of an action unit classified as a 2 compared to no action unit classified as a 0. However, the models struggled differentiating the difference between a moderate appearance of an action unit compared to other classes.

The weighted accuracy, precision, and recall of the three model are reported in Table 1. Weighted accuracy measured the overall accuracy of a model by considering the correctly classified instances in each class as well as the total number of instances in the dataset. This metric is commonly used with a large data imbalance to consider each class equally when determining the model's success. The results in Table 1 showed that the eye model had the highest overall metrics, while the ear model reported the lowest overall metrics. Despite the differing levels of success, all three models reported a weighted accuracy above 0.8.

Table 1: Performance metrics for action unit classification models.

Feature	Weighted Accuracy	Precision	Recall
Eye	0.930	0.91	0.908
Ear	0.808	0.733	0.725
Nose	0.865	0.844	0.836

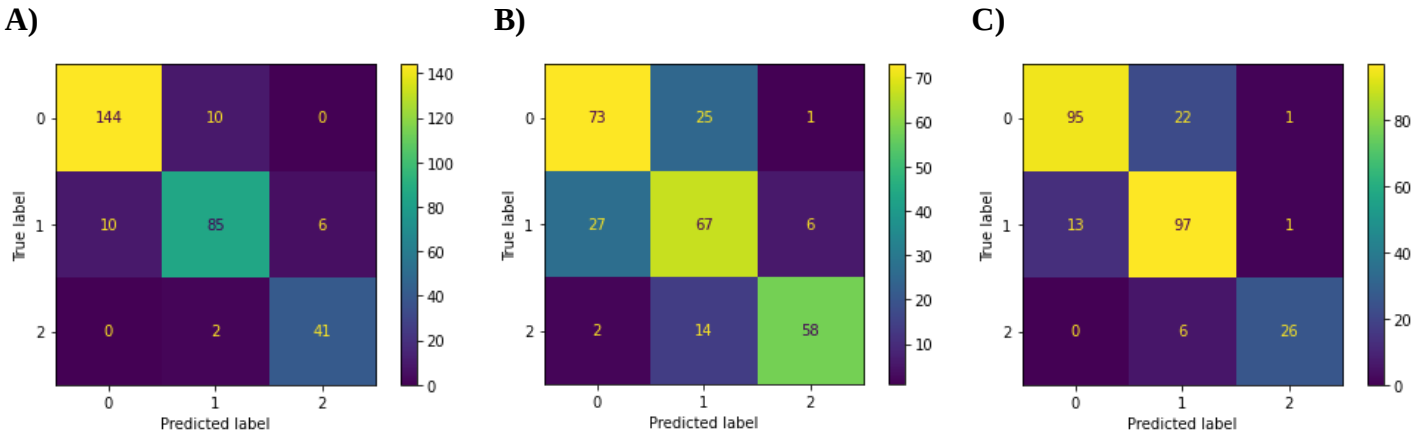

Figure 12: ViT classification confusion matrices: A) eye model; B) ear model; C) nose model.

In order to demonstrate the effectiveness of the ViTs in distinguishing between instances of pain action units and no pain action units, binary classification models were created for 0 and 2 RGS scores. The metrics for these models are displayed in Table 2, while the corresponding confusion matrices are shown in Figure 13. Notably, all models outperformed their three-class counterparts, achieving a weighted accuracy over 0.9. Of the three classifiers, the eye model proved to be the most successful, correctly identifying all but one case of orbital tightening and having no false positives. The nose model also performed well, accurately labeling all but one case of nose flattening and producing a few false positives. The ear model, while not as effective as the other two, still identified over 90% of pain-indicating ear changes and produced few false positives.

Table 2: Performance metrics for binary action unit classification models.

Feature	Weighted Accuracy	Precision	Recall
Eye	0.995	0.985	0.995
Ear	0.910	0.905	0.905
Nose	0.973	0.945	0.970

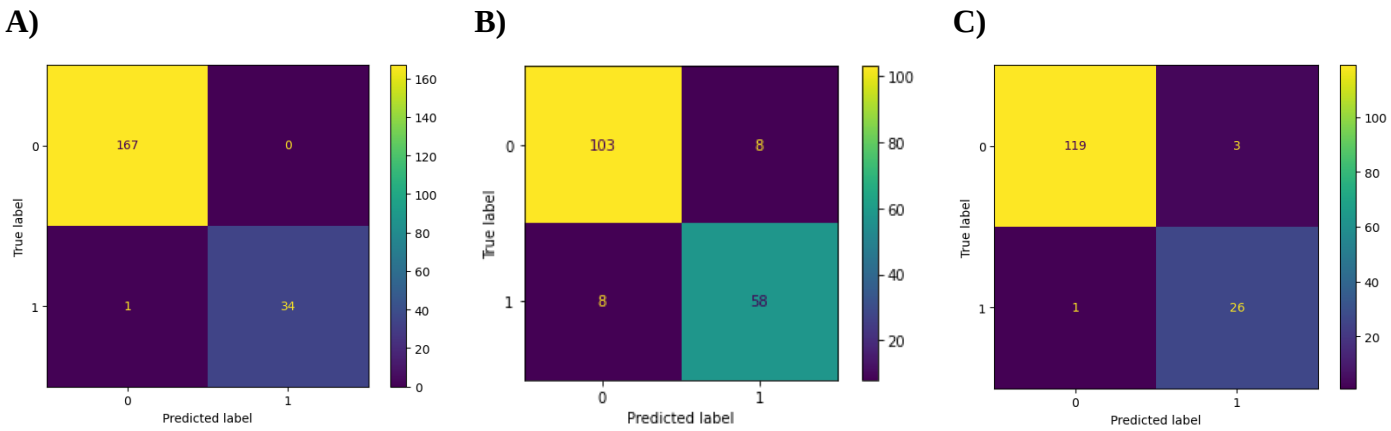

Figure 13: *Binary classification confusion matrices: A) eye model; B) ear model; C) nose model.*

While the metrics of the RGS classification models showed promise, the underlying calculations used to make predictions remain unknown and difficult for researchers to understand. In order to gain further insight, attention heatmaps were generated for each action unit and RGS grade, as shown in Figure 14. These heatmaps provide a visual representation of the weights assigned to each patch of the input image and can help understand which parts of the image the model is used to make decisions.

While examining the attention heatmaps of the eye model, a noticeable pattern appeared, with most prediction being based on the patches surrounding the eye. This suggested that the model considered the shape of the eye, similar to how human graders would evaluate orbital tightening. In contrast, the ear and nose model heatmaps displayed a wider range of attention areas for each prediction, which may be due to the various angles these action units occur. In general, the ear model heatmaps showed attention at the natural fold around the ear canal, while some also demonstrated attention around the ears' edges. This could indicate that the model recognized a decrease in surface area of the ear as the pain level increases. For the nose images, attention is primarily focused on the tip of the nose. Although the features being attended to in these images are diverse, the model seemed to recognize the direction of the nose as an indicator of nose flattening. This can be observed in the attention heatmaps where a line along the direction of the nose can be seen in most cases of elevated RGS scores, reflecting the downward angle of the nose which typically occurs during grimacing.

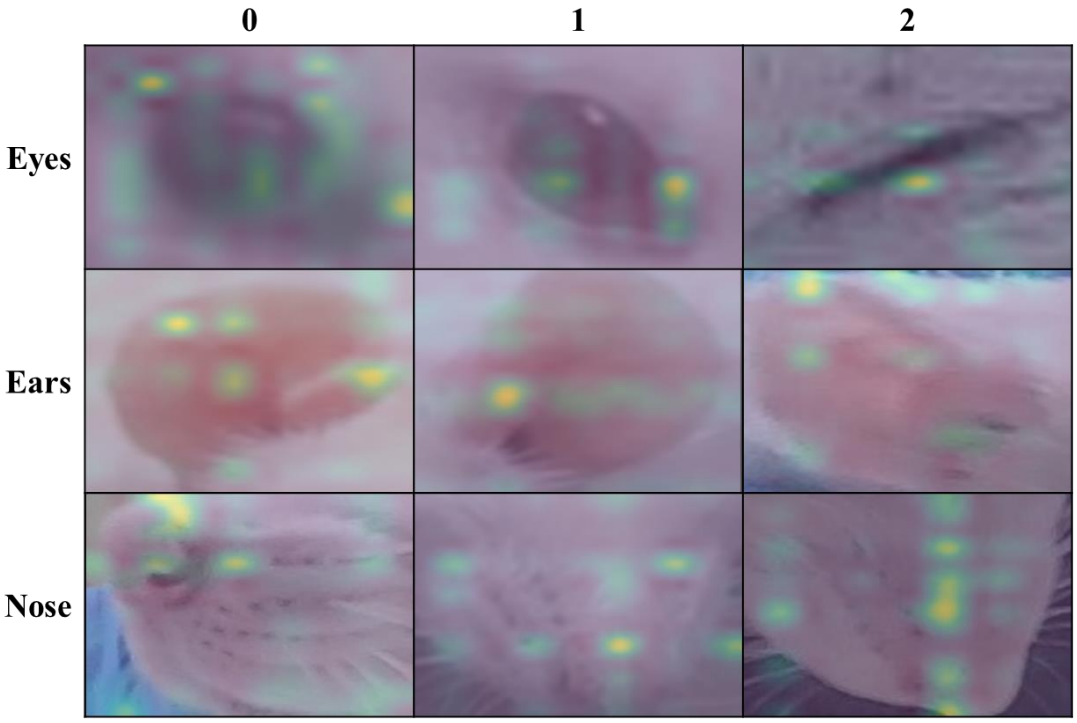

Figure 14: *Attention heatmaps for each action unit at the three RGS classification levels.*

Model Validation

The validation testing of the automated RGS system demonstrated the model's efficacy in detecting pain behaviors. Table 3 provides an overview of the ICC for each action unit between the two graders for the experimental analysis. The ICC for eyes and final RGS score demonstrated a "very good" reliability between graders, and the ICC for ears and nose both demonstrated a "good" reliability [72]. Table 4 presents an overview of the average RGS scores for each validation group and analysis method, while Figure 15 provides a visual representation. A Kruskal-Wallis test was conducted to evaluate the results of the control images in comparison to the 1-day and 1-month images, while a Wilcoxon test was used to compare the 1-day versus the 1-month images. Both the experimental RGS analysis ($p < 0.0001$) and the model analysis ($p = 0.001$) showed that the dataset from the bTBI study on the day of the injury had significantly elevated RGS scores compared to the control images. Additionally, the RGS score notably dropped in the experimental analysis ($p < 0.0001$) and model analysis ($p < 0.0001$) at the 1-month timepoint, indicating a reduction in pain behavior over time. Although both the automated

model and experimental analyses found that the 1-month image dataset had a lower mean RGS score than the initial control group, only the experimental analysis (p = 0.0279) demonstrated statistical significance between the two groups, while the model analysis (p = 0.224) found no significant difference.

Table 3: *Intraclass correlation coefficient calculated for each action unit for the validation set between trained RGS graders.*

Action Unit	ICC
Orbital Tightening	0.952
Ear Changes	0.717
Nose Flattening	0.700
RGS Score	0.882

Table 4: *Mean RGS image score for the control, 1-day, and 1-month groups for the model and experimental analysis.*

Analysis Method	Control	1-Day	1-Month
Model	0.447	0.779	0.328
Experimental	0.320	0.771	0.185

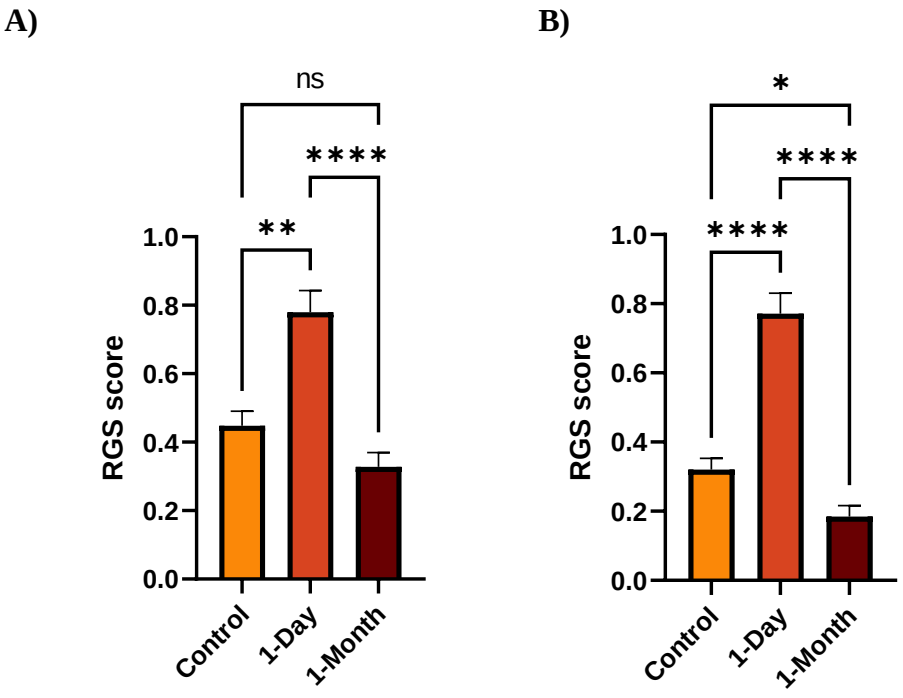

Figure 15: *Analysis of mean RGS score of the control, 1-day timepoint, and 1-month timepoint (ns p > 0.05, * p < 0.05, ** p < 0.01, *** p < 0.001, **** p < 0.0001): A) model analysis; B) experimental analysis.*

A Wilcoxon test was employed to compare the RGS scores obtained from automated model and manual analysis across all three image groups. The results showed no significant difference between the RGS scores obtained from both methods for the control ($p = 0.263$) and 1-day ($p = 0.898$) images, however a significance did appear in the 1-month images ($p = 0.010$). Figure 16 demonstrates that the experimental analysis generated lower RGS scores than the automated RGS system in the control and 1-month groups. Despite the difference between the low RGS groups, the 1-day scores displayed a similar average RGS score between the model and manual grading, indicating that the model was effective in identifying true positives for increased action units. However, it also suggested that the model is more prone to predicting false positives for increased action units. Despite the numerical difference, both analyses demonstrated similar statistical significances when analyzing the 1-day blast group to the other images. Furthermore, an absolute model of ICC shown in Eq. (7) was calculated for each action unit and total RGS score reported in Table 5. Orbital tightening was found to have an ICC greater than 0.9 which indicates a "very good" reliability, while ear changes and nose flattening reported values between 0.61 and 0.80 which indicates "good" reliability [72]. Moreover, the final RGS score calculated by the model was found to have an ICC of 0.818 when compared to the human graders which would be considered "very good" reliability.

Table 5: *Intraclass correlation coefficient calculated for each action unit for the validation set results between experimental graders and model predictions.*

Action Unit	**ICC**
Orbital Tightening	0.930
Ear Changes	0.650
Nose Flattening	0.641
RGS Score	0.818

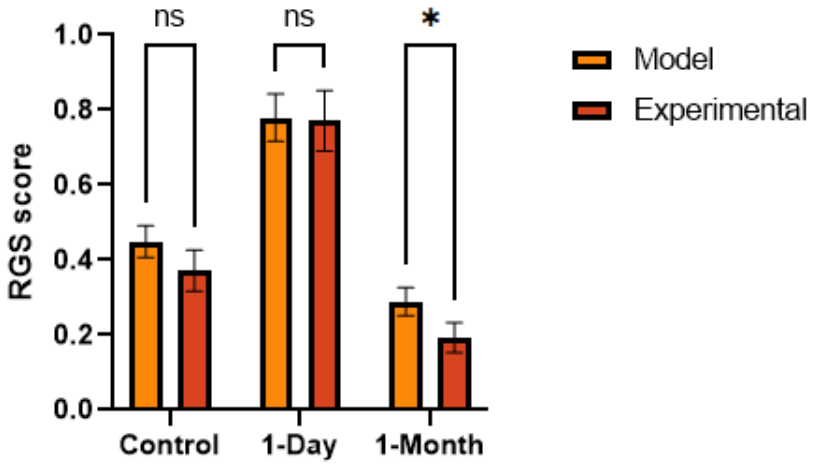

Figure 16: *Analysis of the model and experimental RGS results (ns $p > 0.05$, * $p < 0.05$).*

While the RGS score predictions produced similar results, the difference in efficiency was prominent. The model accomplished grading all 120 images in only 3.2 minutes, while the experimental analysis required 41.5 minutes, illustrated in Figure 17. This demonstrated that the model accomplished grading in $1/14^{th}$ of the time compared to human graders. Moreover, manual grading was performed simultaneously with two graders, resulting in twice the reported man-hours. Furthermore, the graders dedicated hours of training prior to the validation test, highlighting the amount of time required to manually evaluate RGS images.

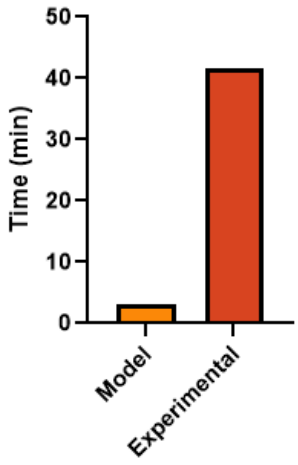

Figure 17: *Total RGS analysis time for the validation set.*

Discussion

An automated RGS system was developed and provided both an efficient and accurate method for RGS image analysis. The system's object detector achieved a validation accuracy of 97.5%, allowing for 117 of the 120 validation images to be graded for RGS score. Furthermore, the action unit classifiers all produced weighted accuracies above 80%. The holistic RGS model performed nearly as well as manual RGS grading but completed the task in a fraction of the time required by human graders. In addition, the model successfully identified statistical differences in RGS scores between a control and a bTBI group, as well as recognizing the reduction in pain behaviors 1-month post injury.

While there are no other automated RGS models published, orbital tightening is used in various animal grimace scales [9, 21-24]. Table 6 depicts the weighted accuracy of machine learning approaches for orbital tightening grading. All of the current methods of orbital tightening grading use convolutional neural networks where a majority of the accuracies are around 0.65 [15, 26, 27, 73]. The DeepMGS system developed in Chiang *et al.* performs significantly better than the other convolutional neural networks producing an accuracy of 0.85 [27]. Nonetheless, despite DeepMGS's improvements over other models, the ViT approach outperformed their accuracy by 8%.

Table 6: *Orbital tightening three-class grimace scale accuracies.*

Study	Species	Weighted Accuracy
Our method	Rats	0.930
Tuttle et al. [15]	Mouse	0.632
Vidal et al. [26]	Mouse	0.650
Chiang et al. [27]	Mouse	0.850
Lencioni et al. [73]	Horse	0.655

Although orbital tightening is easy to compare to currently published models, ear changes and nose flattening are difficult to judge without rat specific models. While ears and the nose do appear in the mouse grimace scale, it considers the ears position rather than ears shape, and the

nose and cheek bulge rather than nose flattening. Despite the different interpretations of these features, DeepMGS reports an ear position accuracy of 0.78 and nose bulge accuracy of 0.81, while our model performed better with an ear change accuracy of 0.808 and nose flattening accuracy of 0.865 [27]. Overall, the ViT approach in the automated RGS system demonstrated a high accuracy in comparison to existing deep learning methods.

The automated RGS system not only exhibited higher accuracies than existing machine learning approaches, but it also demonstrated similar inter-rater reliability measured by ICC to human graders [9, 11, 12, 25]. Table 7 provides an overview of ICC score from various studies, which measures the consistency of ratings between graders. Additionally, Table 7 displays further details on the type of comparison, including whether it entails a single assessment between two graders or multiple comparisons across several individuals. Across all studies, orbital tightening was found to be the most consistent between graders and ear changes and nose flattening were found to have a "good" to "very good" reliability. Notably, studies that included more than two graders generally exhibited higher ICC scores than those with just two graders. In the RGS automated analysis, the ICC scores were found to be consistent with studies with a single pair of graders, and when compared to multiple graders the ICC for ears and nose were slightly below average. Although the automated RGS model did display a strong reliability when compared to human graders, the lower ICC may suggest that it would be best to use the model in conjunction with human graders.

Table 7: *Overview of ICC scores for each action unit across multiple studies.*

Action Unit	Study	Comparison	ICC
Orbital Tightening	Our Model	Single	0.930
	Our Graders	Single	0.952
	Oliver et al. [12]	Single	0.920
	Oliver et al. [12]	Multiple	0.970
	Sotocinal et al. [9]	Multiple	0.960
	Phillips et al. [11]	Multiple	0.920
	Zhang et al. [25]	Multiple	0.840
Ear Changes	Our Model	Single	0.650
	Our Graders	Single	0.717

	Oliver et al. [12]	Single	0.620
	Oliver et al. [12]	Multiple	0.82
	Phillips et al. [11]	Multiple	0.780
	Zhang et al. [25]	Multiple	0.720
Nose Flattening	Our Model	Single	0.641
	Our Graders	Single	0.700
	Oliver et al. [12]	Single	0.620
	Oliver et al. [12]	Multiple	0.83
	Sotocinal et al. [9]	Multiple	0.860
	Phillips et al. [11]	Multiple	0.650
	Zhang et al. [25]	Multiple	0.710

While the automated RGS model presents a promising solution for efficient and accurate RGS image grading, it has limitations that should be addressed. Firstly, even though the model displayed a strong performance for orbital tightening, ear changes, and nose flattening, it does not consider whiskers. Whiskers were part of the original RGS grading criteria, however, without good lighting and high image quality the whisker becomes very difficult to grade. The GoPro HERO8 black used to film the validation set as well as the Von Frey test was capable of producing image quality that could be used in Rodent Face Finder, however it often left the whiskers blurry. Although whiskers have been omitted in many RGS studies, the exclusion of whisker changes within the model will limit its ability to fully implement the RGS.

Secondly, the limited training data may inhibit the model's generalization to other laboratories. While the training data contained datasets from three separate studies and various image qualities, it does not cover all testing scenarios. The automated RGS model was proficient in a validation set with bTBI rats, however images containing visual impairments to the rats such as stitches or wounds may affect the model prediction. Moreover, it is unknown the extent the camera angle, lighting, and image background may affect the YOLOv5 action unit detection.

Furthermore, the ViT action unit classification does not consider the holistic frontal image when grading each action unit. While researchers are advised to consider each action unit separately, they still see the whole image, which may influence the action unit score. The

automated RGS model is limited to grading the action unit area around the eyes, ears, and nose, therefore it cannot account for the complete image context. To make the model receive information analogous to human graders, it may be advantageous to utilize the whole frontal-facing images rather than just the action unit area.

Despite the limitations the automated RGS grader faces, it provides a promising step towards automated pain behavior monitoring. There are many potential avenues for future research and improvement to the current model, with the aim to eventually develop a real-time, comprehensive, and robust rat welfare monitor. The most straightforward approach is the development of a more robust RGS image database. By improving the current database, the model would become more accurate and generalizable to potential RGS images.

Additionally, an improvement to the current Rodent Face Finder should be made. Currently the Rodent Face Finder application uses Haar cascades to detect one eye and one ear as a means to collect frontal-images with minimum blur. While this method works for most images, it does collect images where the nose is not in plain view and requires the video input to run in slow motion to detect objects. One solution for these problems would be implementing the YOLOv5 action unit detector as a frontal image collection system. This system would be capable of detecting all three action units and would have an adjustable confidence threshold. If researchers want to collect clear images, they would be able to set a high confidence threshold and guarantee all three action units are in full view. Furthermore, this method would prevent false negative object detection within the automated RGS system, allowing for a more robust grading system.

Although the system was noticeably faster than human graders, it was still unable to grade images in real-time. The model's average grading speed per image for the validation set was 1.58 seconds. This was due to the image initially running through the YOLOv5 model, then running 3 to 5 subsequent action unit images through their respective ViT model. When considering each model separately they were all capable of running analysis at real-time, but when combined together are substantially slower. One potential solution for this problem would be to eliminate the ViT models and use the YOLOv5 model as both an object detector and RGS classifier. This strategy would be quick enough to run the program on live video and would

consider the holistic image while grading each action, however, it is unknown how effective the model classification would be.

Another solution to developing a real-time automated RGS system would be to use a singular ViT classifier rather than an object detector and action unit specific classifiers. This model would be able to take into account the entire context of an image and produce a singular RGS grade. This strategy was attempted as a binary classifier for the RGS using the current database where images with a RGS less than 0.33 were classified as no pain and images with a RGS greater than 1 were classified as pain. This strategy was attempted due to the success of the binary classifiers for the individual action units displayed in Table 2. This binary model achieved an accuracy of 0.97 and would be capable of running on live video. The 0.97 accuracy was lower than the accuracy for the binary eye classification and comparable to the accuracy of the binary nose classification. However, this model only requires one output that has a high accuracy, whereas the binary classification for individual action units to detect pain would have more opportunities to predict a FN or a TN. Moreover, the model results were analyzed using an attention heatmap illustrated in Figure 18, where the model was seen using the action units area of orbital tightening and ear changes to classify pain images. Even though this model is able to successfully classify pain images and run noticeably quicker, it is still unable to provide granulated action unit scores. Additionally, the model is susceptible changes in background which would affect the generalizability between research groups.

Figure 18: *ViT whole image RGS classification model's attention heatmap for a pain classification [58].*

Nonetheless, the ViT image classifier and attention heatmap may provide interesting research avenues for pain monitoring outside of the RGS. Currently, the RGS considers all four action units equally when making a pain grade. While this works for researchers by making an easily quantifiable measure for pain behavior, it may not be representative of a genuine pain score. For instance, multiple studies have found that orbital tightening is the most robust measure when determining a grimace scale [9, 12]. One study that could be conducted to create a more robust pain score is the replication of the pain assay procedure conducted in Langford *et al.* [21]. However, rather than manually selecting action units, the images could be separated into a control and pain group and classified in a ViT model. After creating the classification model, a post hoc study would be able to use the attention heatmaps and determine what parts of the image are considered when making pain and no pain classifications. This method would not only enable the detection of unrecognized action units in the grimace, but also aid in identifying which current action units are crucial for pain behavior identification and their relative importance.

Conclusion

The development of an automated RGS system for action unit detection and pain behavior classification presents a significant advancement in the field of preclinical pain research. The system was found to be proficient at correctly scoring RGS images in a fraction of the time when compared to human graders. In addition, the model successfully distinguished pain behavior differences between a control, recently injured rats with bTBI, and rats with 1-month of recovery. The findings of this study not only contribute to the standardization of the RGS scores between labs, but also provide a foundation for future research to create a real-time pain monitor and discover new techniques for quantifying pain behaviors. Furthermore, this study presents a promising avenue for pain monitoring beyond the RGS in other animal models. In conclusion, this research represents a significant step towards overcoming the barriers for the utilization of RGS testing and demonstrating the feasibility of real-time grimace monitoring.

References

[1] J. S. Mogil and S. E. Crager, “What should we be measuring in behavioral studies of chronic pain in animals?,” *Pain*, vol. 112, no. 1–2, pp. 12–15, Nov. 2004, doi: 10.1016/J.PAIN.2004.09.028.

[2] M. S. Dawkins, “The Science of Animal Suffering,” *Ethology*, vol. 114, no. 10, pp. 937–945, Oct. 2008, doi: 10.1111/J.1439-0310.2008.01557.X.

[3] D. L. Hickman, J. Johnson, T. H. Vemulapalli, J. R. Crisler, and R. Shepherd, “Commonly Used Animal Models,” *Princ. Anim. Res. Grad. Undergrad. Students*, p. 117, 2017, doi: 10.1016/B978-0-12-802151-4.00007-4.

[4] D. Le Bars, M. Gozariu, and S. W. Cadden, “Animal models of nociception,” *Pharmacol. Rev.*, vol. 53, no. 4, pp. 597–652, 2001.

[5] P. Ekman and W. Friesen, “Facial action coding system: a technique for the measurement of facial movement,” 1978.

[6] A. C. Amanda, “Facial expression of pain: an evolutionary account,” *Behav. Brain Sci.*, vol. 25, no. 4, pp. 439–455, Aug. 2002, doi: 10.1017/S0140525X02000080.

[7] M. D. Samad, N. DIawara, J. L. Bobzien, J. W. Harrington, M. A. Witherow, and K. M. Iftekharuddin, “A Feasibility Study of Autism Behavioral Markers in Spontaneous Facial, Visual, and Hand Movement Response Data,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 353–361, Feb. 2018, doi: 10.1109/TNSRE.2017.2768482.

[8] C. L. Hicks, C. L. Von Baeyer, P. A. Spafford, I. Van Korlaar, and B. Goodenough, “The Faces Pain Scale-Revised: toward a common metric in pediatric pain measurement,” *Pain*, vol. 93, no. 2, pp. 173–183, 2001, doi: 10.1016/S0304-3959(01)00314-1.

[9] S. G. Sotocinal *et al.*, “The Rat Grimace Scale: A partially automated method for quantifying pain in the laboratory rat via facial expressions,” *Mol. Pain*, vol. 7, p. 55, Jul. 2011, doi: 10.1186/1744-8069-7-55.

[10] T. Akintola, C. Raver, P. Studlack, O. Uddin, R. Masri, and A. Keller, “The grimace scale reliably assesses chronic pain in a rodent model of trigeminal neuropathic pain,” *Neurobiol. Pain*, vol. 2, pp. 13–17, Aug. 2017, doi: 10.1016/J.YNPAI.2017.10.001.

[11] B. H. Philips, C. L. Weisshaar, and B. A. Winkelstein, “Use of the Rat Grimace Scale to Evaluate Neuropathic Pain in a Model of Cervical Radiculopathy,” *Comp. Med.*, vol. 67, no. 1, pp. 34–42, Feb. 2017.

[12] V. Oliver, D. De Rantere, R. Ritchie, J. Chisholm, K. G. Hecker, and D. S. J. Pang, “Psychometric Assessment of the Rat Grimace Scale and Development of an Analgesic Intervention Score,” *PLoS One*, vol. 9, no. 5, p. e97882, May 2014, doi: 10.1371/JOURNAL.PONE.0097882.

[13] L. Zhao, “A Facial Expression Recognition Method Using Two-Stream Convolutional Networks in Natural Scenes,” *J. Inf. Process. Syst.*, vol. 17, no. 2, pp. 399–410, Apr. 2021, doi: 10.3745/JIPS.01.0070.

[14] L. P. J. J. Noldus, A. J. Spink, and R. A. J. Tegelenbosch, “EthoVision: A versatile video tracking system for automation of behavioral experiments,” *Behav. Res. Methods, Instruments, Comput.*, vol. 33, no. 3, pp. 398–414, 2001, doi: 10.3758/BF03195394/METRICS.

[15] A. H. Tuttle *et al.*, “A deep neural network to assess spontaneous pain from mouse facial expressions,” *Mol. Pain*, vol. 14, pp. 1–9, 2018, doi: 10.1177/1744806918763658.

[16] M. Feighelstein, I. Shimshoni, L. R. Finka, S. P. L. Luna, D. S. Mills, and A. Zamansky, “Automated recognition of pain in cats,” *Sci. Reports 2022 121*, vol. 12, no. 1, pp. 1–10, Jun. 2022, doi: 10.1038/s41598-022-13348-1.

[17] C. Darwin, *The Expression of the Emotions in Man and Animals*. London: Albemarle, 1872.

[18] K. M. Prkachin, “The consistency of facial expressions of pain: a comparison across modalities,” *Pain*, vol. 51, no. 3, pp. 297–306, Dec. 1992, doi: 10.1016/0304-3959(92)90213-U.

[19] K. J. S. Anand and P. R. Hickey, “Pain and its effects in the human neonate and fetus,” *N. Engl. J. Med.*, vol. 317, no. 21, pp. 1321–1329, Nov. 1987, doi: 10.1056/NEJM198711193172105.

[20] J. S. Mogil, D. S. J. Pang, G. G. Silva Dutra, and C. T. Chambers, “The development and use of facial grimace scales for pain measurement in animals,” *Neurosci. Biobehav. Rev.*, vol. 116, pp. 480–493, Sep. 2020, doi: 10.1016/J.NEUBIOREV.2020.07.013.

[21] D. J. Langford *et al.*, “Coding of facial expressions of pain in the laboratory mouse,” *Nat. Methods 2010 76*, vol. 7, no. 6, pp. 447–449, May 2010, doi: 10.1038/nmeth.1455.

[22] M. C. Evangelista *et al.*, “Facial expressions of pain in cats: the development and validation of a Feline Grimace Scale,” *Sci. Reports 2019 91*, vol. 9, no. 1, pp. 1–11, Dec. 2019, doi: 10.1038/s41598-019-55693-8.

[23] K. B. Gleerup, B. Forkman, C. Lindegaard, and P. H. Andersen, “An equine pain face,” *Vet. Anaesth. Analg.*, vol. 42, no. 1, pp. 103–114, Jan. 2015, doi: 10.1111/VAA.12212.

[24] P. di Giminiani *et al.*, “The assessment of facial expressions in piglets undergoing tail docking and castration: Toward the development of the Piglet Grimace Scale,” *Front. Vet. Sci.*, vol. 3, no. NOV, p. 100, Nov. 2016, doi: 10.3389/FVETS.2016.00100/BIBTEX.

[25] E. Q. Zhang, V. S. Y. Leung, and D. S. J. Pang, “Influence of Rater Training on Inter- and Intrarater Reliability When Using the Rat Grimace Scale,” *J. Am. Assoc. Lab. Anim. Sci.*, vol. 58, no. 2, p. 178, 2019, doi: 10.30802/AALAS-JAALAS-18-000044.

[26] A. Vidal, S. Jha, S. Hassler, T. Price, and C. Busso, “Face detection and grimace scale prediction of white furred mice,” *Mach. Learn. with Appl.*, vol. 8, p. 100312, Jun. 2022, doi: 10.1016/J.MLWA.2022.100312.

[27] C. Y. Chiang, Y. P. Chen, H. R. Tzeng, M. H. Chang, L. C. Chiou, and Y. C. Pei, “Deep Learning-Based Grimace Scoring Is Comparable to Human Scoring in a Mouse Migraine Model,” *J. Pers. Med. 2022, Vol. 12, Page 851*, vol. 12, no. 6, p. 851, May 2022, doi: 10.3390/JPM12060851.

[28] M. A. Hearst, S. T. Dumais,. E Osuna, .J Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, p. 28, Jul. 1998, doi: 10.1109/5254.708428.

[29] Z. Zhang, “Introduction to machine learning: k-nearest neighbors,” *Ann. Transl. Med.*, vol. 4, no. 11, Jun. 2016, doi: 10.21037/ATM.2016.03.37.

[30] I. Rish, “An empirical study of the naive Bayes classifier,” *Intern. Jt. Conf. Artif. Intell.*, vol. 3, no. 22, pp. 41–46, 2001.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[33] “What is a Neural Network? | TIBCO Software.” [Online]. Available: https://www.tibco.com/reference-center/what-is-a-neural-network.

[34] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, Nov. 2015, doi: 10.22214/ijraset.2022.47789.

[35] V. H. Phung and E. J. Rhee, “A Deep Learning Approach for Classification of Cloud Image Patches on Small Datasets,” *J. Inf. Commun. Converg. Eng.*, vol. 16, no. 3, pp. 173–178, 2018, doi: 10.6109/JICCE.2018.16.3.173.

[36] A. Vaswani *et al.*, “Attention is All you Need,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[37] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *ICLR*, Oct. 2020.

[38] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Conference on Computer Vision and Pattern Recognition*, 2001, doi: 10.1109/CVPR.2001.990517.

[39] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[40] O. Déniz, G. Bueno, J. Salido, and F. De La Torre, “Face recognition using Histograms of Oriented Gradients,” *Pattern Recognit. Lett.*, vol. 32, no. 12, pp. 1598–1603, Sep. 2011, doi: 10.1016/J.PATREC.2011.01.004.

[41] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,” in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2013.

[42] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[43] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in neural information processing systems*, 2015.

[44] R. Girshick, “Fast R-CNN,” in *IEEE international conference on computer vision*, 2015, p. 1440-1448.

[45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[46] G. Jocher *et al.*, “ultralytics/yolov5: YOLOv5 SOTA Realtime Instance Segmentation.” Zenodo, 22-Nov-2022, doi: 10.5281/ZENODO.7347926.

[47] C. Li *et al.*, “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications,” *IEEE Conf. Comput. Vis. pattern Recognit.*, Sep. 2022.

[48] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” Jul. 2022.

[49] C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen, “Object detection based on YOLO network,” in *Proceedings of 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference, ITOEC 2018*, 2018, pp. 799–803, doi: 10.1109/ITOEC.2018.8740604.

[50] G. Jocher, A. Chaurasia, and J. Qiu, YOLO by Ultralytics (Version 8.0.0) [Computer Software], 2023, https://github.com/ultralytics/ultralytics

[51] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A Forest Fire Detection System Based on Ensemble Learning," *For. 2021, Vol. 12, Page 217*, vol. 12, no. 2, p. 217, Feb. 2021, doi: 10.3390/F12020217.

[52] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 390–391.

[53] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, vol. 37, no. 9, pp. 1904–1916, doi: 10.1109/TPAMI.2015.2389824.

[55] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[56] J. V. Roughan, H. G. M. J. Bertrand, and H. M. Isles, "Meloxicam prevents COX-2-mediated post-surgical inflammation but not pain following laparotomy in mice," *Eur. J. Pain*, vol. 20, no. 2, pp. 231–240, Feb. 2016, doi: 10.1002/EJP.712.

[57] P. E. Studlack *et al.*, "Blast-induced brain injury in rats leads to transient vestibulomotor deficits and persistent orofacial pain," *Brain Inj.*, vol. 32, no. 13–14, p. 1866, Dec. 2018, doi: 10.1080/02699052.2018.1536282.

[58] A. J. Furman *et al.*, "Cortical 6-9 Hz Oscillation are a Reliable Biomarker of Persistent Pain in Rats," *bioRxiv*, p. 2020.01.02.893289, Jan. 2020, doi: 10.1101/2020.01.02.893289.

[59] D. Pang, "Rat Grimace Scale Rater Training Data," *Harvard Dataverse*. 2018.

[60] W. J. Dixon, “EFFICIENT ANALYSIS OF EXPERIMENTAL OBSERVATIONS,” *Ann. Rev. Pharmacol. Toxicol. 198U*, vol. 20, pp. 441–62, 1980.

[61] Tzutalin, “LabelImg.” [Computer Software] Git, 2015. https://github.com/tzutalin/LabelImg

[62] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” in *ICML*, 2013.

[63] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement.” *arXiv preprint arXiv:1804.02767* (2018).

[64] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[65] T. Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *European Conference on Computer Vision*, 2014, vol. 8693 LNCS, no. PART 5, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48/COVER.

[66] T. Wolf *et al.*, “Transformers: State-of-the-Art Natural Language Processing,” in *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.

[67] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” Jun. 2016.

[68] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, “ImageNet-21K Pretraining for the Masses,” Apr. 2021.

[69] O. Uddin *et al.*, “Chronic pain after blast-induced traumatic brain injury in awake rats,” *Neurobiol. Pain*, vol. 6, p. 100030, Aug. 2019, doi: 10.1016/J.YNPAI.2019.100030.

[70] K. O. McGraw and S. P. Wong, “Forming inferences about some intraclass correlations coefficients",” *Psychol. Methods*, vol. 1, no. 4, pp. 390–390, Dec. 1996, doi: 10.1037/1082-989X.1.4.390.

[71] T. K. Koo and M. Y. Li, “A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research,” *J. Chiropr. Med.*, vol. 15, no. 2, pp. 155–163, Jun. 2016, doi: 10.1016/J.JCM.2016.02.012.

[72] J. Landis and G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[73] G. C. Lencioni, R. V. de Sousa, E. J. de Souza Sardinha, R. R. Corrêa, and A. J. Zanella, “Pain assessment in horses using automatic facial expression recognition through deep learning-based modeling,” *PLoS One*, vol. 16, no. 10, p. e0258672, Oct. 2021, doi: 10.1371/JOURNAL.PONE.0258672.