Real-Time Facial Expression Recognition Using Ensemble Model and YOLO

Jixuan Leng
jleng3@u.rochester.edu

Yuze Wang
ywang349@u.rochester.edu

Chengkai Kang
ckang12@u.rochester.edu

Abstract

Facial expressions display emotions and play a vital role in interpersonal communication. In this paper, We propose a multi-model method to track the real-time emotion of human subjects.

We explore numerous training strategies and hyperparameters on our custom ensemble model, which achieves a 74.48% *accuracy by integrating different techniques. In addition, we combine our ensemble model and a shrunk version of the YOLOv1 network to create a web application using python flask that achieves facial expression recognition in real-time.*

1. Introduction

Facial expressions play a vital role in interpersonal communication. In the early 20th century, Ekman and Friesen defined the term basic emotions which refers to a set of six emotions conveyed by different facial expressions: anger, disgust, fear, happiness, sadness, and surprise [2]. This non-verbal information is key to social interactions between humans. In recent decades, numerous studies have been conducted to develop FER (Facial Expression Recognition) models to automate the task because of their practical applications in security systems, disease diagnosis, law enforcement, and other intelligent human-computer interaction devices. While recognizing posed basic emotions is considered a solved problem [7], distinguishing them accurately under natural conditions remains a challenge.

However, novel deep learning techniques such as Convolutional Neural Networks (CNN) enabled significant progress in building more efficient FER models. Some remarkable results even exceed human performance. In this paper, we aim to better understand these FER models and improve their accuracy by taking different approaches including transfer learning, simulated annealing, data augmentation, label smoothing, data mixup, and ensemble modeling. In addition, we wish to utilize the YOLOv1 algorithm to enable the real-world usage of our model.

2. Related Works

The FER-2013 dataset was created for the facial expression recognition challenge which encourages competitors to design the best FER model. The winner, Yichuan Tang, achieved a 71.162% accuracy by implementing a linear support vector machine (SVM) top layer instead of a softmax layer in the network to learn lower-level features. Additionally, he used the L2-SVM loss function which is differentiable and penalizes errors heavily [10]. The technique was particularly novel and demonstrated outstanding performance at the time.

In a more recent work, Yu and Zhang combined multiple CNN models by minimizing a likelihood loss and a hinge loss function to learn the ensemble weights and applied data augmentation to the dataset to improve performance. In addition to image transformation, they output the response of each test image as an averaged voting of responses from all the perturbed samples [13]. The method reached approximately 72% accuracy on FER-2013 and ranked second in the EmotiW2015 challenge [1].

Research done by Adrian Vulpe-Grigoraşi and Ovidiu Grigore highlighted a method using Random Search Algorithm on a search space defined by discrete values of five hyperparameters to find the optimal values for the neural network [12]. The proposed model had five million parameters and a size of 59MB and achieved a 72.16% accuracy on the FER-2013 testing dataset. As a comparison, a VGG model built by Simonyan and Zisserman had a 72.6% accuracy on the dataset, a total number of parameters ranging from 133 to 144 million, and a size of 528MB [8]. The optimized model produced exceptional results by reducing the number of parameters by approximately 97% and the size by 89%.

Furthermore, Christopher Pramerdorfer and Martin Kampel identified major bottlenecks in CNN-based Facial Expression Recognition, such as basic and shallow architectures and a lack of large generalized datasets. They also demonstrated that overcoming these bottlenecks could lead to a substantial performance increase. For example, they removed the initial CP block of a 34-layer ResNet and created a more narrow network that had 256 feature maps in the final residual group to reduce the number of parameters.

Their modified ResNet achieved an accuracy of 72.4% on the testing dataset [6].

3. Dataset

Among plentiful FER datasets, we chose the FER-2013 dataset to train our emotion classification model and the PASCAL Face dataset to train our face detection model. The two models we are developing aim to solve two distinct problems, thus we need two datasets with different labels and annotations to accommodate them.

3.1. FER-2013

The FER-2013 dataset was created by Pierre Luc Carrier and Aaron Courville using Google image search API and was first proposed in ICML 2013 [3]. The dataset contains 35887 grey-scale 48x48 images which include: 28709 training set images, 3589 public test images, and 3589 private test images. These images are labeled with seven emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. According to Goodfellow *et al*. [3], human accuracy on this dataset was $65 \pm 5\%$. FER-2013 is known to be challenging because it is heavily unbalanced: while there are 7215 images in the Happy class, only 436 images are in the Disgust class, as shown in Figure 1.

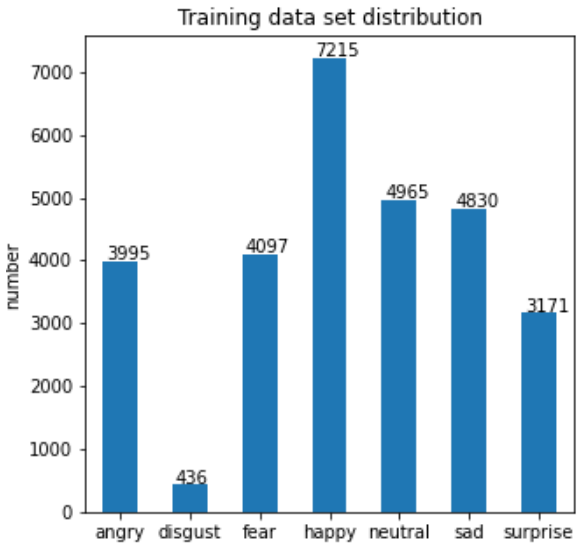


Figure 1. Distribution of the training data set

3.2. PASCAL Face

The PASCAL Face dataset is a sub-dataset within the 2007 PASCAL VOC challenge. The data set contains 450 faces all facing at the camera. The dataset is split in a 10-90 format for testing and training. Ground truth is provided within the dataset annotations in the form of corner coordinates. However, the faces in the dataset vary slightly by their poses, orientations, scales, and lighting conditions. Figure 2 shows an example of our training data and annotation.

```
[280.77896337 507.3845387  279.54572045  48.61817381 618.68752246
  46.15168797 619.92076538 508.61778162]
251 41 621 514
```

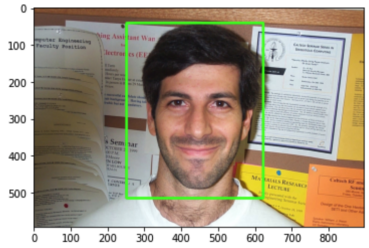


Figure 2. Example of training data and annotation, the dataset is from the Caltech database (http://host.robots.ox.ac.uk/pascal/VOC/databases.html).

4. Emotion Classifier

FER-2013 is a relatively small dataset, thus we decided to apply transfer learning and fine-tuning on various CNN models including PyTorch pre-trained ResNet-50, VGG16, DenseNet-161, Wide ResNet-50, RegNet, and EfficientNet to boost the accuracy of ours. We tested and explored numerous training strategies mainly on ResNet-50.

4.1. Training Parameter

For the final ensemble model, it ran for 150 training epochs with 32 batch size optimizing the cross-entropy loss using stochastic gradient descent.

4.2. Cosine Annealing Learning Rate

One common problem during the model training process is that the gradient descent algorithm gets stuck at a local minimum which causes unwanted convergence. Ilya Loshchilov and Frank Hutter introduced the cosine annealing scheduler [5] to help the model escape the local minimum. The simulated annealing algorithm initializes a large learning rate that decreases rapidly to find a local minimum before increasing again to eventually arrive at the global minimum.

4.3. Data augmentation

The second strategy to boost test accuracy we explored is data augmentation. This technique artificially modifies existing data to improve generalization and effectively prevents overfitting. Our model's accuracy increased to 69.02% after randomly applying conventional data augmen-

tation such as horizontal flip, resized crop, color jitter, vertical and horizontal shift, and random erasing.

To further enlarge the size of training samples, we applied a five-crop transformation before random data augmentation. The method crops the image into four corners and a central piece, which makes the training set five times larger. Test-time augmentation was applied to the validation and testing set. Ten-crop transformation on the validation and testing set allows the model to take the average prediction probability over crops when doing prediction which would increase the accuracy. Figure 3 shows the result of the N-crop and data augmentation on the training and validation data set.

(a) Training data set after data augmentation

(b) validation set with tencrop

Figure 3. Effect of data augmentation

4.4. Mixup and Label Smoothing

As Figure 4 shows, the training data set contained some samples that should not belong to any of the classes. These unexpected samples could lead to a low generalization thereby reducing the performance of our model.

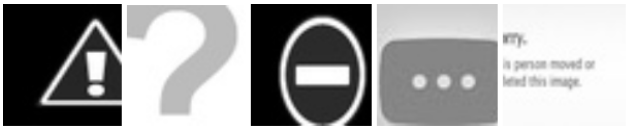


Figure 4. Example of bad samples from training data set

We implemented label smoothing regulation to prevent our model from overfitting on these unexpected samples and giving out predictions extremely confidently. This regularization technique was proposed by Szegedy *et al.* [9]. The idea is to use a noise distribution instead of the one-hot encoded vector to form a new ground truth label distribution and replace the original one in the loss function. Equation (1) shows the math formula for cross-entropy loss with label-smoothing regularization [9].

$$H(q', p) = -\sum_{k=1}^{K} \log p(k) q'(k) = (1-\epsilon) H(q, p) + \epsilon H(u, p) \tag{1}$$

Mixup is a simple and data-agnostic data augmentation routine [14]. It combines two images and their one-hot encoded labels according to some ratio. This establishes a linear relationship between data augmentation and the supervision signal, which reduces the model's memorization of corrupt labels and improves the generalization [14]. The implementation of mixup is shown in Equation (2), where λ follows a beta distribution.

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda) x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda) y_j \end{aligned} \tag{2}$$

After implementing both label smoothing and mixup, the fine-tuned ResNet-50 reached an accuracy of 72.89%.

4.5. Weight Sampling

To alleviate FER2013's class imbalance, we weighted the samples as the inverse of the sample size of each class as shown by Equation (3). However, we did not notice any improvement. We believe this is because the sample variation is too large, and using the inverse of the sample sizes as their weights might negatively influence the learning of classes with larger sample sizes.

$$w = \frac{1}{\text{Number of Samples in Class c}} \tag{3}$$

4.6. Ensemble Model

We applied the same strategies and hyperparameters to train VGG16, DenseNet-161, Wide ResNet-50, RegNet, and EfficientNet. We performed ensembling of six models including our base model by removing the last fully connected layer of each model, stacking them, and creating a new linear classifier for the combined model. The custom ensemble model in the end achieved the highest accuracy of 74.48%.

5. Face Detector

To perform face detection, we used a shrunk version of YOLOv1. Compared with the original model, with its last convolution layer consisting of 4096 layers, our model reduced the number to 496 to save computing power. Moreover, YOLOv1 is trained from the ground up due to the PASCAL dataset's low variation in images and low data points. A learning rate scheduler on a low epoch cycle was also used to achieve the desired accuracy.

5.1. Data Treatment

YOLOv1 uses the Darknet-13 framework as its base, therefore the data provided by Caltech cannot be directly used as inputs. The original annotation contains 8 data points for each face, and each of the 8 points represents the x and y coordinate of the bounding box corner. We

applied Equation (4) below to translate it into Darknet-13 format.

$$
\begin{aligned}
x_{center} &= \frac{(x_{min} + x_{max})}{2} * \frac{1}{\text{image width}} \\
y_{center} &= \frac{(y_{min} + y_{max})}{2} * \frac{1}{\text{image height}} \\
\text{width} &= \frac{(x_{max} - x_{min})}{\text{image width}} \\
\text{height} &= \frac{(y_{max} - y_{min})}{\text{image height}}
\end{aligned} \tag{4}
$$

5.2. Training Parameter

The Yolov1 network ran for combined 300 epochs with 16 batch sizes. The original learning rate is 2e-5 and the weight decay is 0.

5.3. Learning Rate scheduler

Due to the similarity between the data, our model often converges too quickly. We decided to use a plateau learning rate scheduler to combat this problem by adjusting the learning rate between iterations. The plateau scheduler decreases the learning rate by 1e-4 with a patience of 5 epochs.

6. Results

We plotted our experimental results and applied model visualization methods such as Class Activation Maps, confusion matrices, and loss evaluation to better assess the behavior of our models.

6.1. Visualize the model

Figure 5 shows our experiment results. The trend identifies that as we are implementing more and more training strategies and test time augmentation, our model's accuracy on the testing dataset keeps improving.

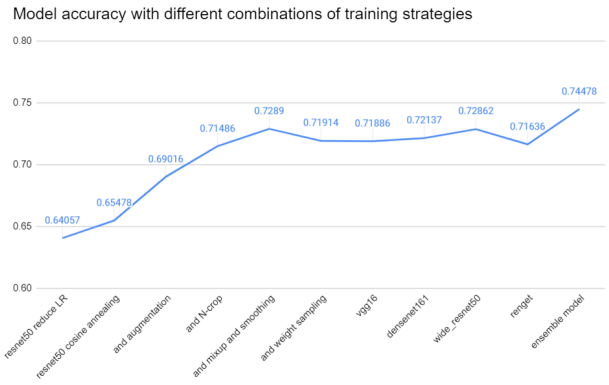


Figure 5. Line graph of experiment results

To better visualize the behaviors of our neural network, we employed Class Activation Map (CAM) [16]. Figure 6 shows the heat maps that highlight class-specific regions of images. Our model had learned to make its predictions mainly based on the lower region of the faces.

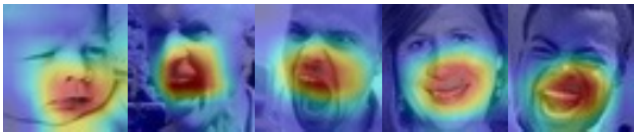


Figure 6. Class activation map of testing data. Region with warmer colors have greater importance in the prediction

We plotted the confusion matrix of our ensemble model on the testing dataset for error analysis. Based on Figure 7, we noticed that there is insufficient test data for disgust. And our model has a high rate of mispredictions for sad, neutral, and fear which is similar to the mispredictions made by humans.

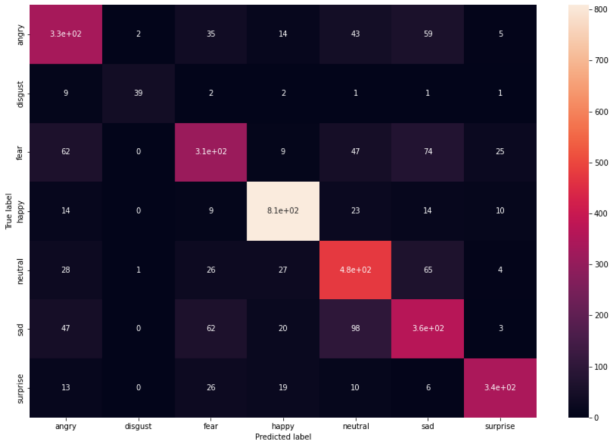


Figure 7. Confusion matrix of our ensemble model

6.2. YOLO result

Due to low variation in learning data, Yolov1 performed relatively well when testing on the dataset. Figure 8 shows that model losses are ranged below 2 percent if excluding outliers.

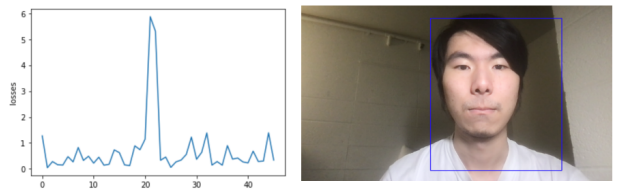


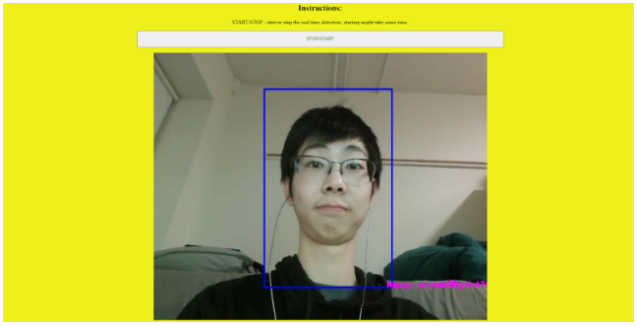
Figure 8. Losses across 45 testing images and images of detected box by trained YOLOv1 model

The model is also robust against faces that did not appear in the training dataset if the picture is consistent in camera angle and lighting.

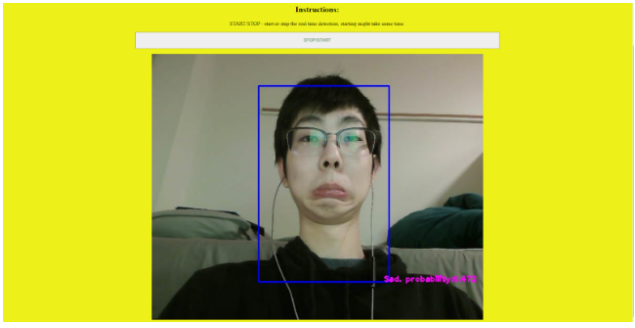
7. Web App

Finally, we wanted to apply our strategy in a real-world environment. By utilizing the python flask framework, we were able to merge our emotion classifier and face detector and deploy it in the form of a web application.

To accommodate the two models, we had to do some image rescaling. First, the detection frame is rescaled to 448 by 448 pixels to fit the darknet-53 standard, which is then fed into the YOLOv1 network to acquire the bounding box of the face. After the bounding box position is extracted, we cropped the frame and apply the necessary image transformation. The frame is then passed into the ensemble model to produce the classification result, which is displayed on the screen with proper prompts as shown in Figure 9.



(a) Happy



(b) Sad

Figure 9. Screenshot of Web app

8. Conclusion

Our goal for this project is to improve the accuracy of the emotion classifier and enable its usage to the real world. We use ResNet-50 as our base model and explore several training strategies. To improve the generalization of our model, avoid model overfitting, and alleviate FER2013's imbalanced and corrupted data, we employ data augmentation, test-time augmentation, data mixup, and label smoothing. Our training strategies significantly increase the base model's accuracy from 64.06% to 72.89%. By ensembling six models, our model's accuracy reaches 74.48%, which is about 3 percent higher than Yichuan Tang, the winner of the Kaggle competition on the FER-2013 dataset with a model of 71.162% accuracy [10]. By applying Class Activation Mapping on correctly predicted data, we observe that our model learns to focus on mouth and nose, important facial features, for emotion predictions.

Additionally, we use a shrunk version of YOLOv1 to realize face detection. The results of YOLOv1 are served as inputs for our custom emotion classifier to perform emotion recognition. By using python flask to develop a web app, we apply our emotion classifier and face detector in real-time.

9. Future Work

In the future, we want to continue to provide modifications to create a more efficient model and generalize it for greater usage.

The two datasets our current model uses are very limited and prone to error. To further increase the accuracy of our emotion classifier, we will explore methods such as ADASYN [4] to balance our dataset according to each class sample's level of difficulty in learning.

We also find that simply ensemble more models increases the accuracy at the cost of a longer prediction time which leads to a frame rate drop in our real-time emotion detector. Thus, in order to make our model more generalized to real-world data and keep a fluent detection at the same time, we will test and try to find a more optimized combination for our ensemble model, and we will collect and add more real-world data to our training set. We will also implement a higher version of the YOLO algorithm to make our face detector more robust.

We are looking forward to expanding our work's capability, which includes being able to capture more micro expressions. This could include multiple inputs such as body movement [15] for the machine learning model to study. In addition, converting the web app into a mobile app makes it more accessible to the public.

We firmly believe that our work has great potential, and the long-term goal of our project is to apply it to various industries. In particular, the adoption of artificial intelligence in daily clinical practices will heavily change the way healthcare providers work. In fact, physicians have been using AI to aid their diagnoses [11]. The employment of FER recognition models will give incredible insight into understanding a patient's emotional states and provide uncharted discoveries of better methods to treat them.

References

[1] Abhinav Dhall, O.V. Ramana Murthy, Roland Goecke, Jyoti Joshi, and Tom Gedeon. Video and image based emotion recognition challenges in the wild. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, Nov. 2015. 1

[2] Paul Ekman and Wallace V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971. 1

[3] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64:59–63, Apr. 2015. 2

[4] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008. 5

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016. 2

[6] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: State of the art, 2016. 2

[7] Evangelos Sariyanidi, Hatice Gunes, and Andrea Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1113–1133, 2015. 1

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 1

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. 3

[10] Yichuan Tang. Deep learning using linear support vector machines, 2013. 1, 5

[11] Bach Tran, Giang Vu, Giang Ha, Quan-Hoang Vuong, Manh-Tung Ho, Thu-Trang Vuong, Viet-Phuong La, Manh-Toan Ho, Kien-Cuong Nghiem, Huong Nguyen, Carl Latkin, Wilson Tam, Ngai-Man Cheung, Hong-Kong Nguyen, Cyrus Ho, and Roger Ho. Global evolution of research in artificial intelligence in health and medicine: A bibliometric study. *Journal of Clinical Medicine*, 8(3):360, Mar. 2019. 5

[12] Adrian Vulpe-Grigoraşi and Ovidiu Grigore. Convolutional neural network hyperparameters optimization for facial emotion recognition. In *2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pages 1–5, 2021. 1

[13] Zhiding Yu and Cha Zhang. Image based static facial expression recognition with multiple deep network learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, Nov. 2015. 1

[14] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017. 3

[15] Guoying Zhao and Xiaobai Li. Automatic micro-expression analysis: Open challenges. *Frontiers in Psychology*, 10, 2019. 5

[16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015. 4