Deep Learning Approaches for Facial Expression Recognition (FER) on the FER2013 Dataset

Britton Chesley, UF ECE

***Abstract*—Facial expression recognition (otherwise known as emotion recognition) is a supervised classification task that aims to classify images of human faces as "happy", "sad", etc. Facial expression recognition (FER) tasks could become instrumental in any situation where machine understanding of human emotion is beneficial. For example, FER is of value in improvements to driver-monitoring systems, medical treatment, robotics, and in human computer interaction systems. As with many supervised image classification tasks, deep learning is one of the best performing solutions to FER tasks. This report shows how select deep convolutional neural network architectures perform on a specific emotion recognition dataset, the FER2013 dataset. The best performing architectures were the deepest CNN's that included regularization to reduce overfitting, even outperforming models created specifically for emotion recognition tasks.**

I. Introduction

Facial expression recognition refers to the supervised classification task of automatically classifying the emotion of a human via an image or video stream of their face. Due to the performance improvements deep learning has provided on image classification tasks in general, the most common approach to FER is deep learning. Emotion recognition usually has images of human faces as input to the algorithm/model, and the output can belong in one of seven classes: "happy", "sad", "angry", "neutral", "disgust", "fear", and "surprise". These are the most basic of human emotions (besides neutral) [1], and are of interest in the FER classification domain. Traditional approaches to FER use manually extracted features such as local binary patterns (LBP), LBP on orthogonal RGB planes, and non-negative matrix factorization [2].

Deep learning approaches are useful in FER due to their automatic feature extraction layers, specifically in convolutional neural networks. Different convolutional approaches will be described in this report, and their benefits/drawbacks in terms of performance will be discused. Basic convolutional neural networks were created and tested on an FER dataset. Certain improvements were made to these models (such as dropout, batch normalization, etc.) to experiment with their performance on a blind validation/test set. Other approaches such as wider feature extraction blocks (DeXpression), and feature localization networks (Deep Emotion) were also tested/validated on the same dataset. These performances are compared to their listed performance in the relevant papers, as well as compared to the performance of the hand-crafted models. Modifications to these architectures were also performed, by means of "widening" both architectures.

II. Description

A. Convolutional Neural Networks

Convolutional neural networks (CNNs) are models that utilize the convolution operation in order learn the features that will feed into a fully connected neural network at the output. Convolutional networks are the de-facto standard for image classification problems, and are the backbone for all of the models in this report. Figure 1 shows the general model architecture for a CNN.

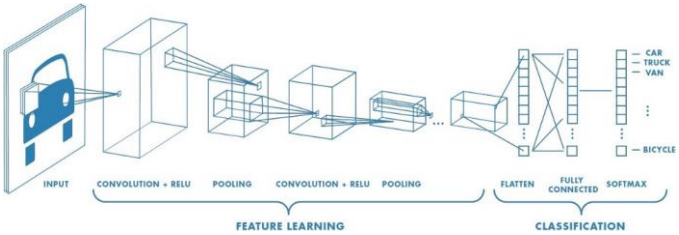

Fig. 1. Convolutional Neural Network Architecture [9]

The first layers in a CNN are the convolutional layers themselves, which utilize a set of learnable filters to extract the important features of an image automatically during training. For example, a filter could learn the feature of detecting a horizontal or vertical line. This line detection can then be fed to further convolutional layers, which can combine the lines to accurately detect the "edge" of a vehicle (as given in the figure). These learned features are called feature maps. Activation functions are used to add a nonlinear component to the network, which allow the network to learn better transformations of the input data to the desired output classes.

Pooling layers are required in a CNN in order to reduce the size of the image fed to the network and the size of the learned feature maps. They also reduce the total number of parameters the model will have, which is good for reducing model complexity. Max pooling layers work by taking the max value of the output of a feature map as the total output for a region of pixels. Most CNNs have many convolutional layers and pooling layers stacked together, to create a "deep" network.

After the final pooling layer, the output "image" is then flattened (vectorized) to be fed into a standard fully connected

neural network. This network combines all of the learned features of the preceding layers and allows for the final classification of the input image. Typically, a one-hot encoded output layer is used, where the number of neurons in the layer equals the number of classes. Each of the output neurons then represent a probability of belonging to that class, and the neuron with the largest value is taken to be the output class label. Stochastic gradient descent and backpropagation can be used to train a CNN just as they are used to train a regular neural network.

B. Datasets

There are many datasets available for FER tasks, which is a driving factor behind why deep learning approaches are successful in this area. The CK dataset [6] contains video sequences of 10 to 60 frames of 123 subjects, with 7 labels corresponding to the 7 basic human emotions. Static methods which use this dataset use the most relevant frames of the videos as input images into their learning pipelines. The Jaffe dataset [7] consists of 213 images of 10 Japanese females showing 6 different kind of human facial expression. The SFEW (Static Facial Expressions in the Wild) dataset [8] is an image dataset created by taking frames from a video-based dynamic dataset (AFEW). These frames were selected based on facial point clustering [2].

The dataset used for all facial expression recognition tasks in this report was the FER 2013 dataset. FER2013 is a widely used FER dataset that consists of 48x48 grayscale images of faces. The faces are centered and occupy the same amount of space in each image [3]. Training was done on a set of 22967 images while validation was performed on a set of 5742 images. The classes of interest as described earlier are: “happy”, “sad”, “angry”, “neutral”, “disgust”, “fear”, and “surprise”. This dataset was constructed using the google search API [2], which allows for a lot of variability in the images. Thus, this dataset is considered to be a “hard” dataset to predict on, and generalization is harder to achieve than on other, more uniform datasets that have entirely frontal-pose images of faces, without any facial occlusions (blockings). The distribution of class labels was not even amongst the 7 classes in the dataset, with “happy” containing the most images at 5783 samples, while “disgust” contained only 345 samples to train on. Sample images from this dataset are provided in Figure 2. Other datasets such as the MMI dataset were considered during the course of this project, but access to these datasets were not granted by the relevant authors. This is the main reason FER2013 was used.

C. Overview of Method

The main goal of my project was to test different CNN architectures and see how the performance varied on the FER2013 dataset. All models were implemented in the PyTorch framework. The first architectures tested were the “baseline” CNN’s that contained only a few layers (both convolutional and fully connected), and were fed the normalized image data to learn from. To improve the performance of these architectures, dropout, batch normalization, increasing model depth, and data augmentation were all experimented with. Most of the training accuracies were at least 75% while the validation accuracies were around 50% for all of these models. The results of these experiments will be discussed in greater detail in the following section.

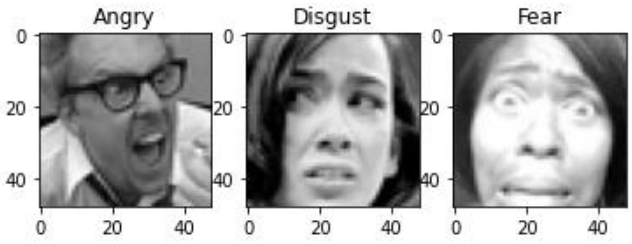

Fig. 2. Sample FER2013 Images from Classes "Angry", "Disgust", and "Fear"

Next, AlexNet and Lenet5 were both trained on FER2013 to see how the standard image processing networks performed on the dataset. These models overfit early in the training process, yet provided decent results similar to the hand-made models. After these two models were trained, an architecture that has wide feature extraction blocks named DeXpression [4] was tested, to see if these feature extraction blocks allowed for better classification accuracy, through learning more discriminative features. The Deep Emotion [5] architecture was then trained to see if using a localization network to detect the most salient regions of an image would improve performance on the tough FER2013 dataset. These results will be discussed in the Evaluation section of this paper.

All models were trained using a batch size of 128 samples, used a one-hot encoding of the output class labels, and used the cross-entropy loss function for training. Adam was used as the optimization method. ReLu was used as the activation function and MaxPool as the pooling layers. All model weights were initialized either by a gaussian random variable with mean 0 and standard deviation 0.05, are with an orthogonal transform. The learning rate was set to 0.001 for the handmade models, although DeXpression and Deep Emotion required a smaller learning rate of 1e-5, to prevent the models from being biased by the uneven class label distribution in a batch.

D. DeXpression Model

Burkert et al. [4] propose an architecture inspired by GoogleNet, which has wide feature extraction blocks that theoretically help learn useful features for classification. The model architecture is shown in Figure 3. The two regions of note are the two squares labelled “FeatEx”. These are the dedicated feature extraction blocks that are the novelty of the model. As can be seen, the feature extraction block consists of two parallel convolutional layers, one acting upon the input data directly and the other acting on the output of another convolutional layer. These layers had many output channels

(filters), and for example convolution 2a has 96 output channels, convolution 2b has 208 output channels, and convolution 2c has 64 output channels [4]. Each convolutional layer in the model also incorporates different filter sizes, to reflect the various scales at which faces can appear [4].

Stacking convolutional layers like layer 2a and 2b allows for learning more hierarchical and combined features, such as learning the outline of a round face, rather than learning the line segments of the face independently. The concatenation blocks towards the end of the FeatEx blocks exists to easily feed the different feature map outputs to the next layers of the model.

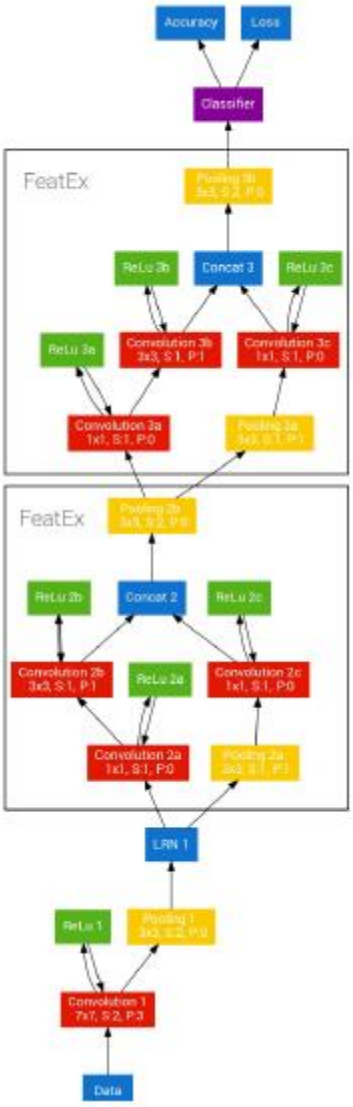

Fig. 3. DeXpression Model Architecture

The DeXpression model architecture was trained on the FER2013 dataset. Modifications were made to this model by means of increasing the width of the feature extraction blocks via more convolutional layers in parallel with the existing feature extraction blocks. The DeXpression network achieved 48.2% validation accuracy at 69.66% training accuracy after 150 epochs. Adding some weight decay regularization mitigated the overtraining some, although the overall validation accuracy was worse and the results are omitted here. Dropout could be implemented to reduce overfitting, although training the model took over four days so not every theorized model architecture could be tested in practice. The authors of this paper report much better test accuracies than are described here, although the DeXpression architecture was not tested on the FER2013 dataset in the original paper. They used more uniform datasets such as MMI and CKP, so the performance achieved on FER2013 is reasonable.

E. Deep Emotion

The Deep Emotion model architecture consists of a backbone CNN and a localization network in parallel with each other. The goal of the localization network is to act as a spatial transformer, in order focus on the important sections of the image [5]. The reasoning behind this is that not all parts of the image are important in detecting a human emotion, so the model should only focus on those most salient regions [5]. Figure 4 shows the architecture of the Deep Emotion Model.

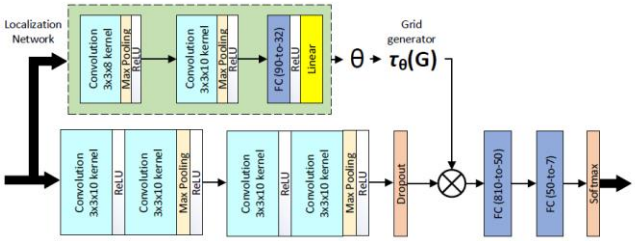

Fig. 4. Deep Emotion Network Architecture

The localization network is a two-layer CNN with max pooling layers and Relu activation functions. The localization network outputs the parameters of the spatial transformation that should be applied to the input [10]. These parameters are then used by a grid generator to create a sampling grid, which is the set of pixels of the input that should be mapped to the output. In this case, these pixels would be closest to the face, zooming in on the eyes, mouth, and nose of the person in the image. The learnable transformation for Deep Emotion is an affine transformation.

This model was trained using the Adam optimizer, with a batch size of 128, L2 regularization on the weights, a learning rate of 0.005 (same as paper), a regularization parameter of 0.001 (not given in paper), and was trained for 500 epochs. The weights were also initialized with a uniform Gaussian distribution with mean 0 and standard deviation 0.05. The Deep Emotion [5] network performed worse than described in its paper on the FER2013 dataset, and yielded 48.56% validation accuracy at 60.90% training accuracy after 340 epochs. The paper reports a 70% testing accuracy on the FER2013 dataset, and this seems reasonable as the model was validated more thoroughly than the model tested here. Also, not all of the hyperparameters were given for training the

model, so this could be another source of error as well.

Model	Validation Accuracy	Training Accuracy	Epoch
A) Baseline net	48.31%	96.41%	20
B) Baseline net with aug.	53.62%	76.10%	210
C) Baseline net with dropout	50.05%	75.49%	60
D) Baseline net with dropout and aug.	55.22%	56.22%	320
E) Baseline deep net with dropout	57.72%	63.48%	280
F) Baseline deep net with batch normalization	53.31%	99.86%	130
G) Deepest net with bn, augmentation	61.37%	87.125%	460
H) AlexNet	47.20%	60.92%	20
I) LeNet5	50.30%	63.98%	10
J) DeXpression	48.20%	69.66%	150
K) DeXpression with wider featEx block	31.45%	32.07%	450
L) Deep Emotion	52.12%	66.26%	200
M) Deep Emotion Wide	49.56%	65.03%	60

Table 1. Summary of Results

III. Evaluation

The main way models were evaluated was by comparing validation accuracies. Note that the validation and training sets used were the same across all models. Many models were tested, but the main results are summarized in Table 1. A summary of the models trained will be given in the following sections. The associated learning curves for the handmade models are shown in Figure 5, and note the subplots are labelled with same letter on the top left as is the heading where their discussion lies below (and the bold letter in the table). The learning curves for AlexNet and LeNet5 models are shown in Figure 6. Figure 7 shows the learning curves for the Deep Emotion and DeXpression architectures.

A. Baseline Net

The baseline net was a simple CNN that had two convolutional layers with 10 and 32 output channels, then followed by two fully connected layers of that had output sizes of 100 and 7, respectively. 48.31% validation accuracy was achieved, although the model started to overfit after only 20 epochs of training.

B. Baseline net with data augmentation

Data augmentation was then added to act as a regularization strategy to reduce overfitting and improve model generalization. The transformations applied to the input images was a random choice of the following: 10-degree random affine transformation, random horizontal flip with 50% probability, random resized crop of 48x48 images, and a random rotation of up to 10 degrees. This improved validation accuracy to 53.62% at 76.10% training accuracy after 210 epochs

C. Baseline net with dropout

This architecture was the same as in A, but with dropout layers added after each convolutional layer with a probability of 0.4. This improved validation accuracy to 50.05% and reduced the overfitting seen in just the baseline net.

D. Baseline net with dropout and data augmentation

B and C were combined to train the shallow "baseline" model with both dropout and data augmentation. The augmentation is the same as in B. This achieved good generalization with 55.22% validation accuracy and 56.22% training accuracy after 320 epochs of training. This generalized the best out of all models as the training and validation (test) accuracies are the most similar, although the validation accuracy is not as high as in G.

E. Baseline deep net with dropout

The model depth was then increased to allow for more complex mappings to be learned. The model depth was increased to four convolutional layers, with output channels of size 10, 32, 64, and 128 followed by fully connected layers of output size 100 and 7 (the number of classes). Dropout was used on the convolutional layers to prevent overfitting, with a probability of 0.4. This increased validation accuracy to 57.72% while training accuracy was at 63.48% after 280 epochs.

F. Baseline deep net with batch normalization

Next, the same deeper architecture with 4 convolutional layers was trained with batch normalization layers instead of dropout. Batch normalization helps prevent the input distributions to each layer of the network from changing wildly by the transformation applied by the weights and activation function. This is known as internal covariate shift [11]. 53.31% validation accuracy and 99.86% (overfit) training accuracy were achieved after 130 epochs of training.

G. Deepest net with batch normalization, data augmentation

This was the deepest model tested, and included five convolutional layers of output channel width 10, 32, 64, 128, and 128. Each convolutional layer was followed by a batch normalization layer. A fully connected network then followed, with 128 input neurons, two hidden layers with 64 and 32 neurons, and followed by an output layer of 7 neurons. This achieved the best validation accuracy of 61.37% at 87.13% training accuracy after 460 epochs of training.

H. AlexNet

AlexNet needed to be modified slightly to accept 48x48 images, and this was achieved by removing a max pool layer and modifying the fully connected layers such that the output layer has 7 classes. AlexNet achieved decent generalization, with 47.20% validation accuracy and 60.92% training accuracy after 20 epochs. The model began to overfit after 20 epochs, which can be seen in the learning curve in Figure 6. Using weight decay was implemented as an experiment to reduce overfitting, although this prevented the model from achieving better than the listed validation accuracy, so the results are omitted.

I. LeNet5

The LeNet5 architecture needed some modifications to be trained on the FER2013 dataset. The first fully connected layer needed less neurons, and the output layer was changed to learn 7 classes instead of 10. 50.30% validation accuracy was achieved at 63.98% training accuracy after 30 epochs. After 30 epochs, the model overfit substantially. Weight decay was implemented to reduce overfitting, but similar results as in AlexNet were achieved, where the model couldn't reach a validation accuracy that was larger than what is listed in Table 1. The learning curve for this model is presented in Figure 6.

J. DeXpression

This model needed to be modified similarly to Alexnet in order to accept 48x48 images. The DeXpression model architecture achieved 48.2% validation accuracy at 69.66% training accuracy after 150 epochs. This is somewhat different than the paper describes, although no regularization or data augmentation were mentioned in the paper. Figure 7 shows the learning curve for this model.

K. Wider DeXpression

The DeXpression architecture was modified to include two more convolutional layers in parallel to the existing architecture in both FeatEx1 and FeatEx2 as seen in Figure 3. Specifically, a convolutional layer with 128 output channels was added in parallel with Conv2b, and a convolutional layer with 128 output channels was added in parallel with Conv3c. This was theorized to improve model generalization by increasing the number of features learned by the model. However, as the Table 1 shows, this model architecture actually performed the worse out of any model trained, with 31.45% validation accuracy and 32.07% training accuracy. It seem that after 350 epochs the loss value started oscillating, and no further improvements were made. This might be mitigated by testing other optimization strategies besides Adam, but time did not permit more testing.

L. Deep Emotion

The Deep Emotion model architecture performed relatively well with 52.12% validation accuracy at 66.26% training accuracy after 200 epochs of training. The dropout layer after the first fully connected layer helped prevent the model from overfitting, and the spatial transformer seemed to be helping the model generalize on the validation set by focusing on the most important regions of the image.

M. Wider Deep Emotion

The deep emotion architecture was modified by creating another spatial transformer network in parallel with the existing spatial transformer, then concatenating the two outputs from the networks and feeding this into the fully connected layer. The unmodified spatial network consisted of a two-layer CNN with 8 and 10 output channels. The additional spatial transformer is a two-layer CNN that had 24 and 32 output channels. The filter size of the parallel spatial transformer was also decreased from kernel sizes of 7 and 5 in the original architecture to 5 and 3 in the new architecture. The idea was to see if the two spatial transformers learned different transformations that focused on the two most important regions in the image. However, the model performance actually decreased when the second spatial transformer was used, achieving 49.56% accuracy on the validation set and 65.03% training accuracy after 60 epochs. This is most likely due to the concatenation of both outputs of the spatial transformer confusing the fully connected layer it feeds into. One transformation turned out to be better than two parallel transformations for emotion recognition tasks.

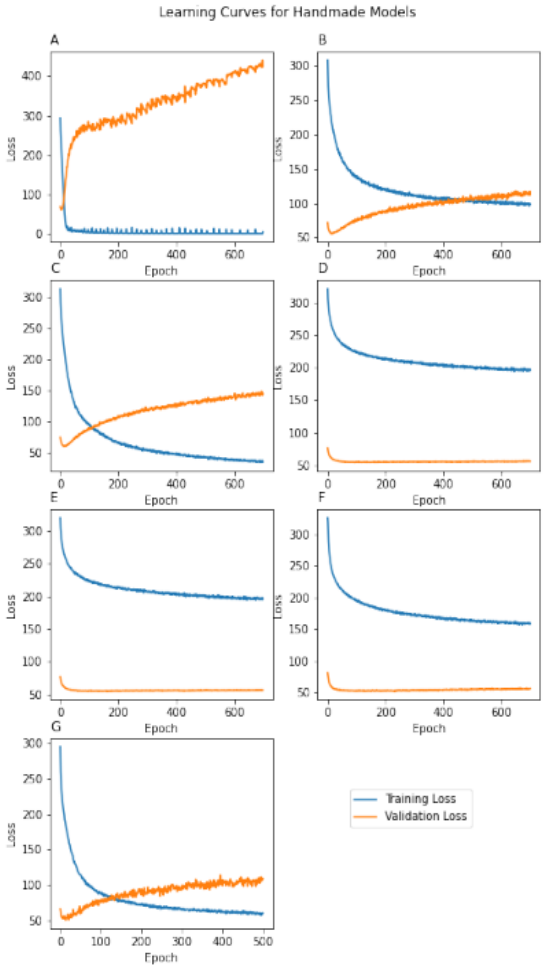

Fig. 5. Handmade model learning curves

N. Discussion of Results

The learning curves in Figure 5 show how the training of the handmade models proceeded. As can be seen in A and F, these models showed lots of overtraining in terms of accuracy and in terms of the validation loss, which skyrockets as the training loss decreases. Models with data augmentation and/or dropout show learning curves with validation errors that don't rise as dramatically, as in D. Surprisingly, the best performing model was one of the "typical' deep CNN architectures, specifically the one described in G. The two special models of DeXpression and Deep Emotion failed to achieve validation accuracies that were similar to what was listed in the paper. If even deeper models were tried, while implementing data augmentation, then even better results could hypothetically be achieved. This is left for future work.

One thing to note is that the DeXpression and Deep Emotion models tended toward becoming biased to "happy" as it has the most samples in the training set. In order to prevent this, the batch size parameter and learning rate parameter had to be adjusted very minutely for proper training to occur.

The confusion matrices in Figures 8, 9, and 10 show the predicted outputs vs the true labels for the deepest model, DeXpression and Deep Emotion. For all models, since class 3 (happy) has the most samples, the models prioritize learning this class. Since class 1, "disgust", has almost 5 times less data than the other classes, the models rarely (if ever) predict this class. However, it seems the main source of error for the deepest network is switching up classes 4 and 5, which correspond to "neutral" and "sad". This is understandable, as many of the facial action units are similar across these two classes. The same error is evident in Deep Emotion, although more error is evident throughout classes 2-5. Since DeXpression had less than 50% accuracy, the resulting confusion matrix in Figure 10 has many misclassifications. Interestingly, it seems DeXpression classifies many samples as "angry", misclassifying almost every other class besides "surprise". "Angry", "surprise", and "neutral" were classified most correctly, since these images tend to have the most varied facial expressions. Based on these results, deeper models have better learning capabilities than wider models such as DeXpression.

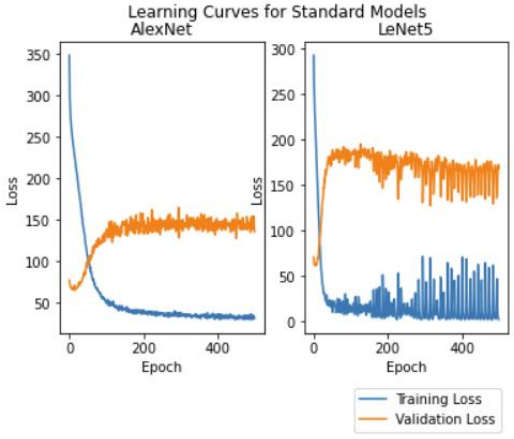

Fig. 6. Learning curves for Alexnet and Lenet5

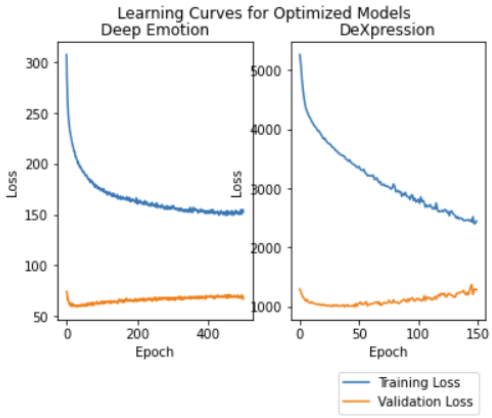

Fig 7. Learning curves for Deep Emotion and DeXpression

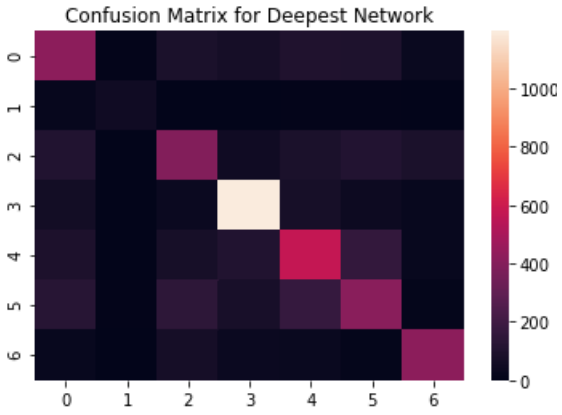

Fig 8. Confusion Matrix for Deepest Network, Model G

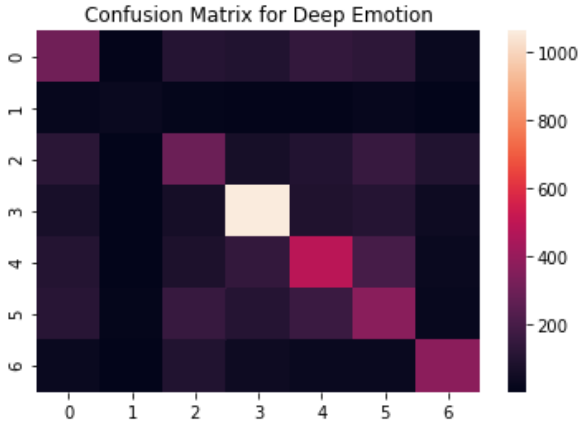

Fig 9. Confusion Matrix for Deep Emotion

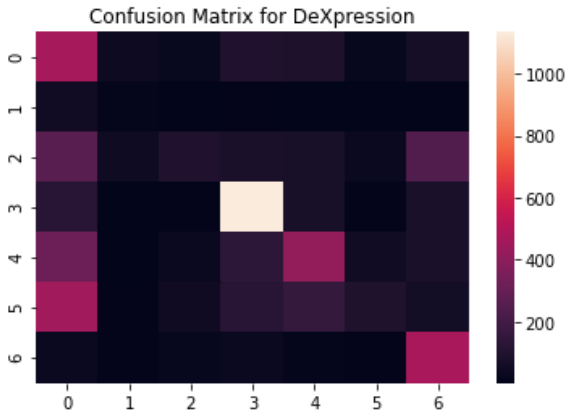

Figure 10. Confusion matrix for DeXpression

O. *Notes on Model Training*

Although the results displayed in Table 1 aren't the best (CNN typically do better on these types of image processing tasks), I did ensure that proper training occurred. This proved to be a difficult task for the Deep Emotion and DeXpression network, and many different iterations of training ensued. Although the FER optimized models of DeXpression and Deep Emotion didn't perform as well as predicted, training these models was rather difficult to prevent the loss function from oscillating. Also, some of these models took days to train, so the overall project took a lot more time than anticipated. Training all of the listed models and obtaining results from them easily surpassed the 30 hours of work mentioned on the course website.

IV. Related Work

Most FER datasets are created with images of people with no obstructions in the image around the facial region. These obstructions could include glasses, masks, or anything that blocks part of the face in the image. Also, these images are usually taken by cameras that look directly at the subject, without any variation in the pose of the person making the facial expression. These assumptions make training CNN's much easier, with better accuracies, but are unrealistic for employing these models "in the wild". [12] aims to create pose and occlusion robust architectures that allow for facial expression recognition on images that include occlusion of the face and different poses of the facial region.

Wang et al. [12] propose using a Region Attention Network to capture the importance of facial regions, which is composed of a self-attention module, a relation module, and a feature extraction module. This theoretically allows the model to weight the more important regions of an image more heavily during classification. The authors also suggest using a novel loss function, which encourages a high attention weight for the most important region of an image [12]. The main benefit of using this architecture is that it automatically chooses which region of the image is most important in determining the overall facial expression, similar to how the Deep Emotion model's spatial transformer works. However, if data augmentation is used with random cropping, this should have a similar effect on model performance, although the random transformations are bound to pick both good regions and bad regions of the image to crop. The main drawback behind this method is that two new datasets needed to be created, as both pose labels and occlusion labels were needed by the architecture to train properly. This architecture achieved 83.63% test accuracy [12] on the same FER2013 dataset used in this paper, which is a much better result than shown here.

Sun et al. [13] propose using multiple features learned by a CNN, a region-based CNN, and a bag of words model based on multi-scale dense SIFT features to train an ensemble of support vector machines. The output of these support vector machines is given to their novel "fusion network" which estimates the input conditional class probability densities. This achieved good results on in the wild datasets, displaying 51.08% blind test set accuracy on the SFEW dataset, but was not tested on FER2013.

Ensemble methods are popular for emotion recognition tasks, and Lee et. Al. [14] proposes using many CNN's in parallel, trained on different subsets and/or transformations of the input data, then exponentially weighting the decisions of each CNN to achieve a final classification result. Each CNN has different filter sizes and depths, to vary the features learned amongst the ensembled learners.

Generative models have also been proposed to address the pose estimation problem. In [15] a generative architecture is used to generate frontal face outputs from typical profile face input images. This is an instance of multitask learning, where both the frontal image and the class label are outputs from the model. Specialized loss functions were used to implement each learning task.

V. Conclusions

To conclude, the best performing architecture was the deepest CNN model that included batch normalization and data augmentation. Surprisingly, this outperformed both the DeXpression model and Deep Emotion model, although the performance is not as high as the accuracy listed in the Deep Emotion paper. This shows how powerful deep CNN architectures can be, and their advantages over wider networks such as DeXpression. Deep Emotion failed to achieve its listed performance in its paper, but did perform relatively well compared to some of the other shallower models.

Many PyTorch implementation skills were gained throughout the project. Also, debugging the training of different models with different learning rates and batch sizes was a notable skill gained during the implementation of this project. The field of facial expression recognition is broad, and many different implementations exist for this task. Learning about the varied approaches, and implementing code used for real research was both interesting and rewarding.

References

[1] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." Journal of personality and social psychology, vol. 17, no. 2, pp. 124–129, 1971

[2] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," *arXiv.org*, 22-Oct-2018. [Online]. Available: https://arxiv.org/abs/1804.08348. [Accessed: 25-Apr-2021].

[3] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee et al., "Challenges in representation learning: A report on three machine learning contests," in International Conference on Neural Information Processing. Springer, 2013, pp. 117–124.

[4] P. Burkert, F. Trier, M. Z. Afzal, A. Dengel, and M. Liwicki, "DeXpression: Deep Convolutional Neural Network for Expression Recognition," *arXiv.org*, 17-Aug-2016. [Online]. Available: https://arxiv.org/abs/1509.05371. [Accessed: 25-Apr-2021].

[5] S. Minaee and A. Abdolrashidi, "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network," *arXiv.org*, 04-Feb-2019. [Online]. Available: https://arxiv.org/abs/1902.01019. [Accessed: 25-Apr-2021]. B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[6] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. IEEE, 2010, pp. 94–101.

[7] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on. IEEE, 1998, pp. 200–205.

[8] "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011, pp. 2106–2112.

[9] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way," *Medium*, 17-Dec-2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 25-Apr-2021].

[10] Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems, 2015.

[11] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv.org*, 02-Mar-2015. [Online]. Available: https://arxiv.org/abs/1502.03167. [Accessed: 25-Apr-2021].

[12] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition," *arXiv.org*, 05-Sep-2019. [Online]. Available: https://arxiv.org/abs/1905.04075. [Accessed: 25-Apr-2021].

[13] B. Sun, L. Li, G. Zhou, X. Wu, J. He, L. Yu, D. Li, and Q. Wei, "Combining multimodal features within a fusion network for emotion recognition in the wild," in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015, pp. 497–502.

[14] B.-K. Kim, H. Lee, J. Roh, and S.-Y. Lee, "Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition," in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015, pp. 427–434.

[15] Y.-H. Lai and S.-H. Lai, "Emotion-preserving representation learning via generative adversarial network for multi-view facial expression recognition," in Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on. IEEE, 2018, pp. 263–270.