CATEGORIZING FACIAL EXPRESSIONS

A PREPRINT

Rodolfo Cho Cubarrubia Jr
School of Data Science
DS6050 - Group 2
University of Virginia
Charlottesville, VA 22904
rcc7u@virginia.edu

Shrikant Tribhuvannath Mishra
School of Data Science
DS6050 - Group 2
University of Virginia
Charlottesville, VA 22904
stm5ne@virginia.edu

Daniel Endre Kiss
School of Data Science
DS6050 - Group 2
University of Virginia
Charlottesville, VA 22904
vfq4xe@virginia.edu

December 3, 2023

ABSTRACT

As facial recognition software has advanced, the scope of its applications has expanded from basic characteristics identification to more nuanced tasks such as expression retrieval, photo album summaries and head-shot selections. We focus on the task of human emotion detection, aiming to build a neural network model capable of accurately classifying emotions depicted in human facial images. To create this neural network, we explore encodings and embeddings through autoencoders to create a latent space that captures human emotion. Our goal is to create a reliable emotional latent space through solely unlabeled training data while also experimenting with semi-supervised pre-training, where the encoder will be trained on a small bit of labeled training data and mostly with unlabeled training data. We use the Google FECS dataset for training and the AffectNet dataset for testing.

Keywords emotion detection · facial expression · emotion recognition · FACS · DS6050 · Group 2

1 Motivation

As facial recognition software has improved, the questions of interest have evolved from identifying basic characteristics into more sophisticated applications like expression retrieval, photo album summaries, head-shot selections and the like. Our project focuses on the task of human emotion detection. We aim to create a neural net model that will receive an image input of a human face and accurately classify the emotion shown in that image.

Our motivations stem from two different areas of theory and application, respectively. Our first motivation concerns the supervised nature of most existing facial recognition datasets and the neural nets trained on them. Most publicly available datasets used for facial recognition focus on the discrete emotions defined by the Facial Action Coding System (FACS) [1]. These defined emotional categories do not always capture the true subtleties in human emotions. Think of facial expressions that show a happy cry, or a pained smile, or a nervous laugh. These types of emotions do not always fit nicely into discrete bins. There may even be emotions that have not yet been classified.

Instead of this approach, we will use our chosen dataset, the Google Facial Expression Comparison Dataset (FEC) [2], to encode a representation of human emotions across a continuous, multi-dimensional space. This allows for a representation that is closer to how humans process and evaluate emotions in other humans — wading through ambiguity to make a judgment call. With this encoding, we will be able to use classifying techniques like K-means to allow the model to learn and cluster its own emotional space, which will then be used as classifier training for the model. By choosing an appropriate encoding technique, we hope to create an accurate representation of the true emotional space, which will result in our model handling vague facial expressions and similarly difficult scenarios without relying on human determined labels (but we will still have to implement discrete categories for output).

Our second motivation explores the applications of this model. Ideally (assuming a properly chosen embedding), this model would be able to take unlabeled images, predict and then assign an emotion class to that image. This would be a massive benefit to libraries, archives, museums, photo-wire services and any organization that deals with large sets of images. Archivists and photographers could upload entire photo sets and have the model predict emotion tags (and potentially other tags with a more generalized model) with minimal to zero human oversight needed. If the tags are not already assigned, the model could match search queries in real time (although this is likely an inefficient use) or go through already uploaded material to retroactively assign tags.

Consider a large photo-wire service like Getty Images. When a media outlet wants to run a story, their art department will either dig through their own archives or (more often) license a photo from a service like Getty. These images are carefully chosen to match the tone and message of the story. For example, if *Billboard* intends to run a negative story on Taylor Swift, the art department will want an image that matches that vibe and will then search Getty for images with terms like “angry”, “upset”, “mad” and other similarly unflattering words. Getty could use our model to better handle their existing archives as well as easily handle the constant stream of new photographs that they upload daily, greatly improving their photo service. We envision further use cases where our model, or one similar, could help alleviate the massive loads of photo data that human workers must manually tag themselves, often in large AI data centers in countries like India and the Philippines.

2 Literature Review

Understanding the true nature of human facial expression, and subsequently emotion, has challenged researchers since at least the 1970’s. Paul Ekman at the University of California, San Francisco explored this idea in 1977 [3], questioning if our judgements of human emotions are accurate, if universal facial expressions of emotions can be determined and how emotions form and appear on a human face. Although Ekman found "consistent and conclusive evidence that accurate judgments of facial expression can be made", he encountered much greater difficulty in determining universal facial expressions, largely due to anthropological and cultural factors that lead to differences in interpretation. But Ekman ultimately found a baseline among humans, noting that the appearance of the face for primary emotions was common across cultures (though he also did note a couple of ways that facial expressions differ across cultures).

While facial expression recognition made significant progress in the last decade, building accurate and efficient models still presents significant challenges. In January 2019, Raviteja Vemulapalli and Aseem Agarwala from Google published a study [1] introducing and using the same dataset used for training in this project — the FEC. Vemulapalli and Agarwala were able to create an embedding of facial expressions in a continuous, multi-dimensional space. This embedding was built using the NN2 version of the pre-trained FaceNet with an output of a 7x7 feature map with 1024 channels. This output is then processed by a DenseNet consisting of a 1x1 convolution layer followed by a Dense block. The subsequent output from the DenseNet is then passed to a 7x7 average pooling layer, a fully connected layer and finally an embedding layer. Vemulapalli and Agarwala combined their proposed embedding with a K-Nearest Neighbor classifier to be used as an emotion classifier and applied it to the AffectNet dataset to test (which we will also use for testing purposes) [4] [5].

Taking another route, the team of Jean-Benoit Delbrouck, Noé Tits, Mathilde Brousmiche and Stéphane Dupont from the University of Mons created a transformer-based joint-encoding for both emotion recognition and sentiment analysis in June 2020 [6]. This team created a monomodal and a multimodel transformer encoding; both encodings appropriately relied on attention mechanisms and Feed-Forward Neural Networks. Visual features were extracted from video using a pre-trained CNN. For the multimodel transformer, the team added a dedicated transformer for each modality they worked with. They also proposed further ideas, including a joint-encoding, a modular co-attention mechanism and a glimpse layer at each block.

Other non-neural net efforts have also been made in regards to facial expression recognition. Behnaz Nojavanasghar, Tadas Baltrušaitis, Charles E. Hughes and Louis-Philippe Morency explored emotion recognition in children in 2016 [7]. The group introduced the EmoReact dataset containing 1102 videos annotated for 17 states. After visual and acoustic behavior analysis, the group created different features corresponding to actions like head rotation and lip stretching. Using these engineered features, the group performed classification using classical machine learning techniques like linear and radial basis function kernel SVM classifiers.

Despite these advances, there are still large gaps in our understanding of emotion. Human emotions are inherently complex and are expressed through multiple modalities, including speech and body language; we often struggle with accurately identifying emotional states solely by analyzing facial expressions. There are also additional ongoing challenges related to addressing and mitigating biases that lead to disparities in accurately recognizing facial expressions among different demographic groups, such as age, race, ethnicity, and gender [8]. Some of the most significant controversies troubling data scientists revolve around bias and privacy concerns. Facial emotion recognition represents a notable advancement in the field, as it not only identifies individuals but also discerns their emotional states, enabling systems to respond accordingly even if that response might be considered unfair or wrong by human standards. How can we effectively collect, mitigate bias, and employ facial emotion recognition in an ethical and secure manner? Once deployed, there is the persisting existential question of what impact it will have on our society when our emotional states are recognized everywhere we go [9].

3 EDA and Data Processing

The FEC dataset meant for training originally comprises over 70,000 URLs that point to images. The preprocessing involved a multi-step procedure of retrieving each unlabeled image, detecting the face by utilizing the provided coordinates and subsequently cropping and resizing the resulting image to showcase only the face at a dimension of 200x200 pixels. In a similar manner, the labeled AffectNet dataset, to be used for testing, underwent preprocessing to align its image size with those in the training dataset, ensuring consistency at 200x200 pixels. The AffectNet dataset contains about 15,000 images grouped into eight different sub-directories by the emotions happy, sad, fear, surprise, contempt, disgust, anger and neutral.

Although images were sized at 200x200 pixels and placed to ensure a close crop of the subject's face, the full colored images still contained non-facial features like hair, clothing, backgrounds and different lighting. Faces were also at various angles and not consistently front facing, resulting in images with different ratios of facial to non-facial space. This motivated us to experiment with different preprocessing methods, including setting the images to grayscale and cropping closer to the faces. With both alternate methods, we hoped the models would gravitate towards facial expression features, and be less "distracted" by other features and variations. In the experiment described in the following section, we trained each model on all three versions of our datasets — base preprocessing (RGB), grayscale and close cropping — and compared results. 25 pixels were cropped on either side of each image, resulting in 150x150 pixel images.

Finally, we determined that 25k images was the optimal number that could be pulled from the original Google FEC data as the training set, to balance between the limits of our compute resources and RAM requirements, while still using enough data to train a effective model. The full testing set was utilized.

4 Models and Experiments

To view our code, please visit https://github.com/danielkissUVA/DS6050_project/tree/main. Please note that permissions may need to be granted.

4.1 Process

Using autoencoders, we created embeddings representing the latent emotional space. Using the encoder portion of the model, each autoencoder resulted in its own embedding. K-means algorithm was applied on these embeddings to identify eight clusters to match the eight emotions present in the AffectNet dataset. Next, by using t-SNE to visualize the clusters, and randomly sampling images from each cluster, we manually assigned each emotion to its best matched cluster. To test, we used the same process to encode the test data, and then evaluated it using the K-means algorithm and our manually assigned clusters. Models were judged by accuracy on the AffectNet dataset unless otherwise noted.

To understand the effectiveness of an unsupervised approach to emotion classification, we devised an experiment where each autoencoder model can be compared against a "semi-supervised" and supervised model with the same preprocessing applied. This will help determine whether an unsupervised approach can compare against more proven methods to image classification. The test accuracy metrics are presented in the Results section.

4.2 Unsupervised

Our attempts to create a latent emotional space using unsupervised methods focused on two different stacked autoencoders: one with fully connected dense layers and one with convolution layers. The reason for this is to understand whether either distinct architecture has an advantage in image reconstruction and classification over the other.

The former model is built upon a combined encoder and decoder (as are all our autoencoders). The autoencoder receives images that are 200x200 pixels with three channels for the input. The encoder is a six layer sequential model containing a flatten layer for that input, four dense layers with varying hidden units with SELU (Scaled Exponential Linear Unit) activation, and a dense output layer with 100 units and SELU activation. The dense layers become progressively smaller, from 2000 hidden units to 1000 hidden units to 500 hidden units to 200 hidden units. Please see Figure 1 for a visualization of the stacked autoencoder's encoder portion.

The decoder is a six layer sequential model containing a flatten layer for the input (the encoder's output), four dense layers with varying hidden units with SELU activation, a dense layer with 200x200x3 hidden units with SELU activation, and an output layer that recreates the 200x200 pixels with three channels shape from the autoencoder's input. The decoder is constructed in a mirrored fashion to the encoder, with the dense layers stacked in reverse order to properly reconstruct the input from the encoder into an image.

We arrived at this structure by experimenting with different amounts of layers and different sizes of layers. Less layers with less hidden units led to more abstract image reconstructions while a thicker model with denser layers led to more distinct image reconstructions. But improvements did not simply continue as we increased the amounts and sizes of layers; training quickly became unstable on our university's computing system. This structure provided the best validation performance we could find under the circumstances. Experiments with sparsity regularizers to create sparse stacked autoencoders consistently resulted in worse validation performance. We measured loss through binary crossentropy and used an Adam optimizer with the learning rate set to 0.0001 after experimentation and trained for 10 epochs.

The second autoencoder described earlier primarily consists of 2D convolution layers. The motivation behind selecting convolution layers stems from the goal for this model, which is to recognize the visual patterns and edges within an image that comprise a facial expression. The encoder for this model is a ten layer sequential model, containing a combination of a rescaling input layer, six convolution layers, and three 2D max pooling layers. Each convolution layer consists of the SELU activation function and a kernel size of 3. Additionally, the number of output filters increases from a range of 32 to 128 deeper into the architecture. The encoder architecture for this model can be seen in Figure 2.

The associated decoder in this autoencoder is a seven layer sequential model. Similar to the previous architecture, it is intended to mirror the encoder, with six transposed convolution layers with the same number of filters for each respective convolution layer in the encoder. The initial "deconvolution" layer has an input shape depending on the shape of the encoder's output. Furthermore, while most layers also have the SELU, the final transposed layer has the sigmoid activation function. The overall final layer of the decoder reshapes the output to the same dimensions as the input images.

Also similar to the other previous model, this model's final architecture resulted from extensive experimentation of different number of layers and hyperparameters. The final number of layers, filters, kernel sizes, strides, as well as overall combination of layers were selected to achieve an appropriate balance between training time, performance, and keeping memory usage under our compute resources' limits. Larger filter values caused frequent GPU memory issues. Another bottleneck in terms of memory usage and computational time came from the process of fitting t-SNE and K-means clustering to the generated embeddings from the encoder.

Finally, there were a few slight differences in model parameters depending on the preprocessing applied to the data. If the grayscale images were being tested, the input and output layers of the autoencoder were adjusted to account for the single filter. Since both t-SNE and K-means require a limited amount of dimensions, the generated embeddings were reshaped before fitting to the two algorithms. For the models that were trained on cropped images specifically, the combination of a reduced input size and three filters led us to adjust both stride and pool size parameters in the convolution layers. After extensive testing, it was determined that this one model could only handle a limited training set of 10k images, rather than the 25k applied to every other model.

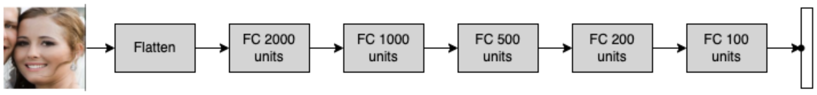

Figure 1: Encoder architecture from stacked autoencoder

4.3 Unsupervised Pre-trained

To compare our unsupervised autoencoders with other approaches, we implemented an unsupervised pre-trained model. This "semi-supervised" approach trains a stacked autoencoder on the entire unlabeled training set. Then, after freezing the encoding layers and combining with a few Dense layers, we use this feedforward model to train on a small amount of labeled data taken from the test dataset. For this experiment, 20% of the data was used for training.

This semi-supervised autoencoder takes a convolutional approach for the encoder and decoder portions. As such, it almost has the same architecture as the convolutional autoencoder described earlier. The main difference between two arises from the fact that this model does not generate latent space embeddings to be fit by t-SNE or K-means. As

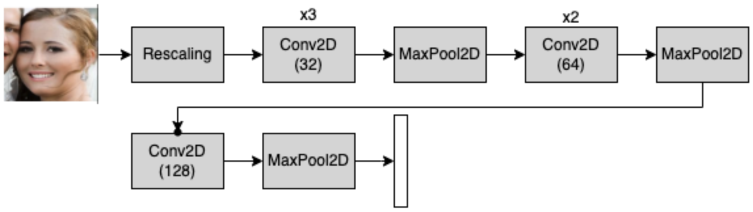

Figure 2: Encoder architecture from stacked convolutional autoencoder

such, we had more flexibility in expanding the complexity of the network, by increasing the filter sizes for each of our convolution and deconvolution layers. Compared to our previous model, which had filter sizes ranging from 32 to 128, this model's primary parameters ranged from 128 to 512 for its encoder layers.

For the initial training on unlabeled images, we measured loss through binary crossentropy and used an Adamax optimizer with the learning rate set to 0.0001 after experimentation and trained for 5 epochs.

After training on the unlabeled images, we froze the encoder's layers and added in 3 dense layers of decreasing neurons (256 to 8), and performed supervised training on the Affectnet test dataset, which consisted of the 8 emotion labels. Unlike earlier, this model was trained on the Adam optimizer with a learning rate of 5e-5. As this is a classification problem with multiple classes, categorical cross-entropy was selected as the loss function.

The motivation behind this model arises from a potential compromise behind our research interest, unsupervised classification, and the proven baseline, supervised classification. In application, perhaps it may not be possible to generate correct classification for facial expressions without explicit labels. It may also require too much of a time investment to label images from large datasets meticulously. The unsupervised pre-trained approach may offer a balance between the two, by requiring a smaller portion of the investment in labeling data than the latter but perhaps also offering greater results than the former.

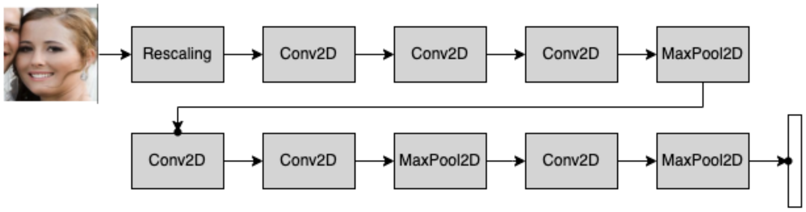

Figure 3: Semi-supervised encoder architecture

4.4 Supervised

Considering the unlabeled nature of our FEC dataset, the supervised model required a different approach. To train and evaluate this model, we split the test AffectNet dataset into train and test splits and only used this dataset for building and evaluating this model. Unlike our other models, this model is not an autoencoder but rather a deep convolutional neural net. This model does not learn an embedding like our autoencoders but instead predicts emotions based on likelihood.

Our supervised model starts with a rescaling layer, followed by seven identical 2D convolution layers with 64 filters, kernel size of three, same padding and ReLU activation. Those are followed by a 2D max pooling layer. Next, another seven identical 2D convolutional layers of the same earlier structure are followed by another 2D max pooling layer. Three more identical 2D convolutional layers of the same earlier structure come next, followed by a flatten layer. Finally, a dense layer with 256 hidden units and SELU activation leads to a dense layer with 8 hidden units and softmax activation to produce our output. We measured loss through categorical cross-entropy and used an Adam optimizer with the learning rate set to 5e-5 after experimentation, and trained for 5 epochs. Please see Figure 4 for the architecture of this model.

Essentially, this model is the baseline to compare the unsupervised clustering approach against. This type of architecture has demonstrated good results on similar problems in the past, so there is a level of expectation that we have for the supervised model.

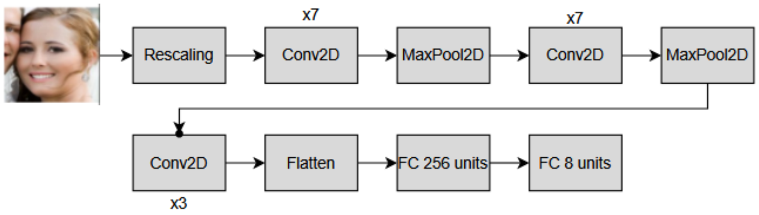

Figure 4: Supervised model architecture

5 Results

Figure 5 shows a t-SNE visualization of the embedding created by the stacked autoencoder when trained on the RGB (base) version of the FEC dataset, while Figure 6 shows image reconstructions made by that autoencoder. Stacked autoencoders with narrower dense layers and shallower architectures resulted in sparser image reconstructions than this. It is interesting to note that while the image reconstructions tend to capture the tone of the image, and the background and head shapes of the subject, the actual expression is considerably more distorted.

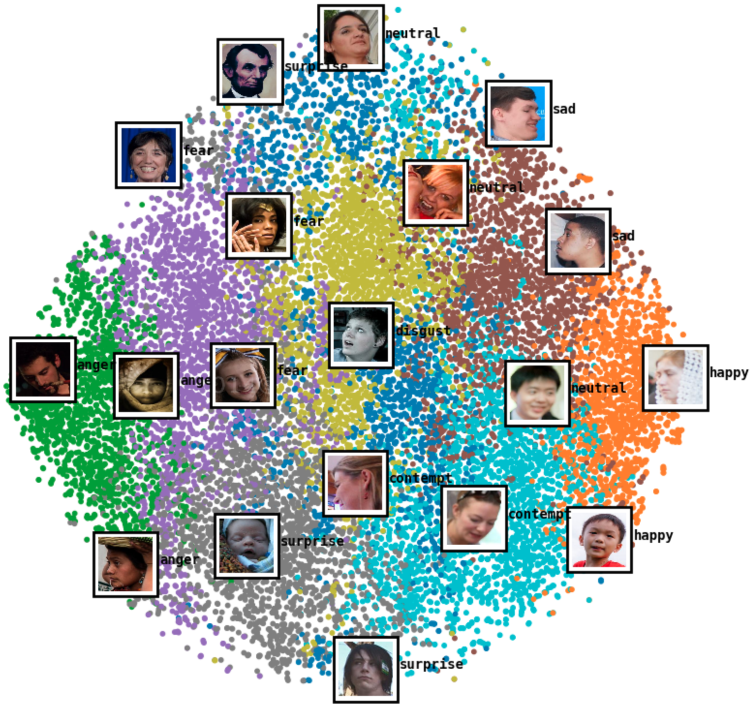

Figure 5: t-SNE visualization of the embedding created by the stacked autoencoder trained on RGB (base) data

Figure 7 below shows a t-SNE visualization of the embedding created by the stacked convolutional autoencoder when trained on the RGB (base) version of the FEC dataset, while Figure 8 shows image reconstructions made by that autoencoder. While achieved on different preprocessing, it is immediately apparent that t-SNE visualizes the high dimensional embeddings in an interesting shape in two dimensions. Additionally, while the images in Figure 8 are in grayscale, it is also noticeable that the addition of convolution layers has caused the reconstructions to appear much closer to the original images, with facial expressions remaining almost intact.

Table 1 represents test performance metrics by accuracy for three distinct approaches—unsupervised, semi-supervised, and supervised—across three types of images: RGB (base dataset), grayscale, and cropped. Notably, the supervised approach consistently outperforms the others, achieving the highest validation accuracy across all image types, with cropped images yielding the best results. This suggests that supervised learning, particularly when applied to carefully

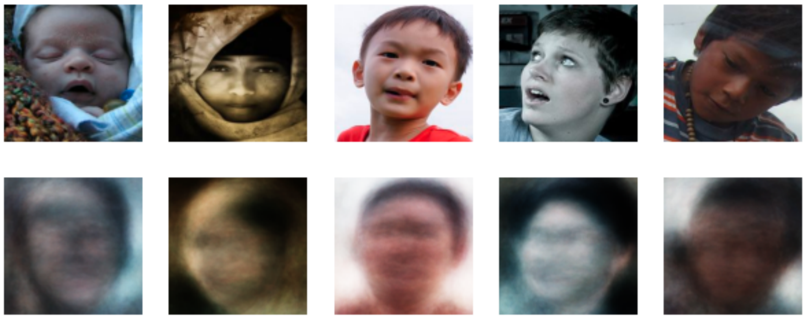

Figure 6: Image reconstructions created by the stacked autoencoder trained on RGB (base) data

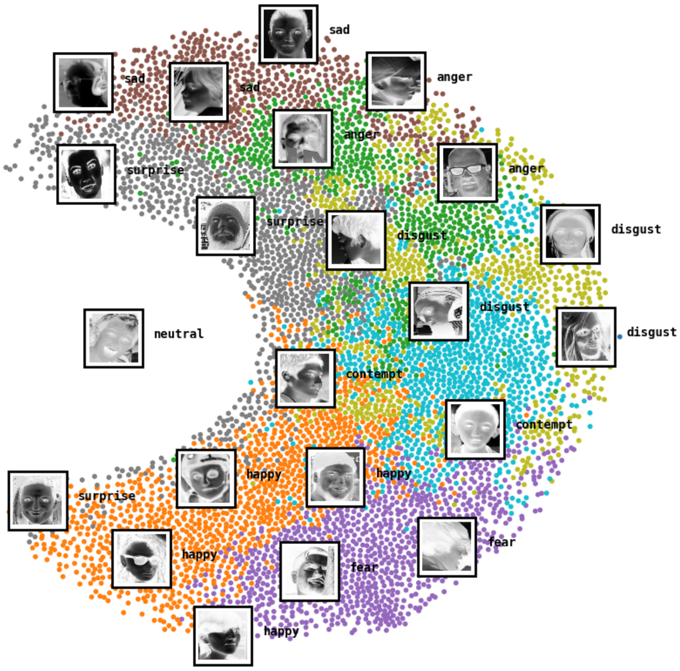

Figure 7: t-SNE visualization of the embedding created by the convolutional stacked autoencoder trained on grayscale data

cropped images, is still the most effective strategy in the given context. On the other hand, the unsupervised approach exhibits lower overall performance, with grayscale images showing slightly better results than RGB and cropped. The semi-supervised approach falls in between, demonstrating significant improvement over unsupervised learning, and its performance varies across image types, being highest for grayscale.

	RGB	Grayscale	Cropped
Stacked autoencoder	.1226	.1253	.1184
Stacked convolutional autoencoder	.1098	.1217	.1777
Semi-supervised	.3431	.4358	.4150
Supervised	.5615	.5751	.5789

Table 1: Model Performance

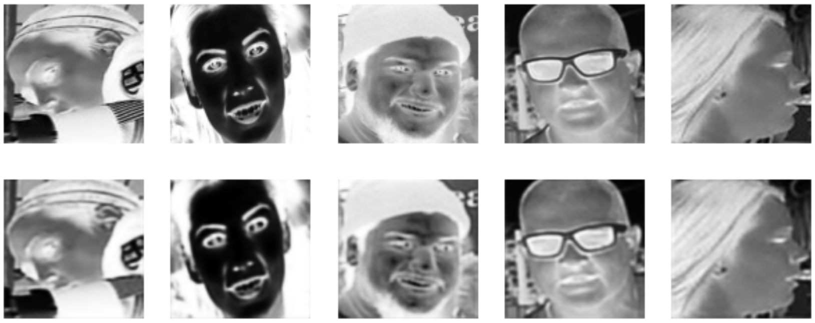

Figure 8: Image reconstructions created by the convolutional stacked autoencoder trained on grayscale data

6 Conclusions

Given the superior performance of the supervised approach, particularly with cropped images, training on labeled data can still be considered the best approach to this problem. The performance on cropped images for the supervised model specifically indicates that removing the 25 pixels from each side of an image did help the model focus on facial expressions or key emotional cues. The moderate validation accuracy in the high 0.5's may indicate that the classes themselves do not have separate clear boundaries that a model can recognizes. The Affectnet dataset has 8 labels, and many of them can certainly have similar facial emotions. For example, "anger", "disgust", and "contempt" all could have similar facial expressions, and even a human may not be able to accurately identify the nuances that define each of the 8 classes. To delve deeper into this, a potential next step for this research question could be to group the positive and negative emotions for a binary classification problem.

Perhaps the biggest challenge we faced lies in the black-box nature of neural nets and our difficulties immediately directing them to the features we want them to learn, particularly in a unsupervised capacity. In classical machine learning, our ability to feature engineer allows us to direct a model towards our desired data representations; the EmoReact project used engineered features like head tilt and mouth curvature to reflect aspects of emotion, allowing their model to directly focus on facial features [7]. When training, our autoencoders would often zone in on other features in the images like clothing, hairlines, backgrounds and lighting. This motivated our alternate preprocessing methods of grayscale and close cropping; we attempted to "nudge" our models towards focusing on facial features in hopes the neural nets would be less "distracted" by the other potential features. Labeled training data seemed to be the only way to "nudge" our models towards facial features specifically. This appears to be a limitation on unsupervised methods with neural nets generally and certainly was a heavy one in regards to our attempts at emotion recognition.

Finally, it appears that the unsupervised pre-trained approach performed relatively well, considering that the supervised model was averaging around in the 0.5's in each category. Certainly better than random chance (which the purely unsupervised models' results were on par with), this result does provide confidence that for such tasks, unsupervised training can still be used, albeit with further time investment in labeling a portion of the dataset. For massive labeling tasks on unlabeled datasets, with the other changes discussed in the next section, this could be the preferred approach.

7 Next Steps

Integrating multiple models into an ensemble, considering the unique attributes of each approach, could potentially enhance the robustness and versatility of emotion recognition in photographs. Exploring transfer learning with large, pre-trained models and fine-tuning these models on emotion-specific datasets could further optimize their effectiveness in real-world scenarios, contributing to more accurate and nuanced emotion labeling for diverse photographic content. Further preprocessing methods, such as other image normalization methods and grouping the more nuanced labels into a more definitive class could lead to improved results.

Optimizing the model for scalability and resource efficiency enables the application to deploy and run seamlessly across various platforms. Utilizing cloud services such as Google Cloud, Amazon AWS, or Microsoft Azure offers substantial scaling capabilities and resources, enhancing overall performance. Deploying the model on mobile devices poses a distinct challenge, necessitating the minimization of deployment size while maintaining optimal performance. Figure 9 shows an example of how these types of models could be deployed. We envision applications like an in-app integration that automatically generates labels when a photo is taken or uploaded or a service that can evaluate archived images and assign them labels automatically.

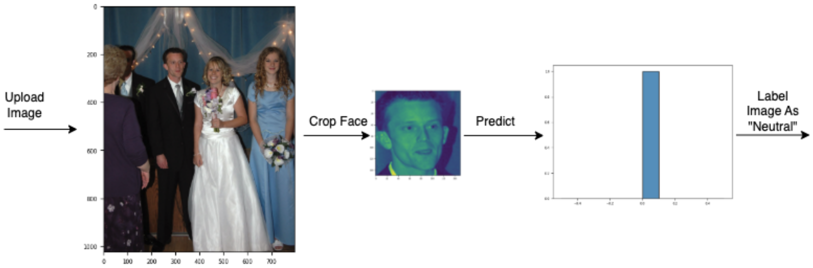

Figure 9: Example application

8 Team Contributions

This was a true collaborative effort, with each member contributing significantly to each aspect of the endeavor. The model training, experimentation, and evaluation were equally divided amongst all three members.

References

[1] Vemulapalli, R., & Agarwala, A. *A Compact Embedding for Facial Expression Similarity*. *arXiv preprint arXiv:1811.11283* (2019, Jan 9) https://arxiv.org/pdf/1811.11283.pdf

[2] Vemulapalli, R., & Agarwala, A. *Google Facial Expression Comparison Dataset*, 2018. https://research.google/resources/datasets/google-facial-expression/

[3] Ekman, Paul *Facial Expression*, 1977 https://www.paulekman.com/wp-content/uploads/2013/07/Facial-Expression.pdf

[4] Mollahosseini, A., Hasani, B., & Mahoor, M.H. *AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild arXiv preprint arXiv:1708.03985* (2017, October 9) https://arxiv.org/pdf/1708.03985.pdf

[5] Mollahosseini, A., Hasani, B., Mahoor, M.H., & Segal, N. *Facial Expressions Training Data (AffectNet)* (2023, January 16) https://www.kaggle.com/datasets/noamsegal/affectnet-training-data

[6] Delbrouck, J.B., Tits, N., Brousmiche, M., & Dupont, S. *A Transformer-based joint-encoding for Emotion Recognition and Sentiment Analysis* (2020, June 29) https://arxiv.org/pdf/2006.15955.pdf

[7] Nojavanasghari, B., Baltrušaitis, T., Hughes, C.E., & Morency, L.P. *EmoReact: A Multimodal Approach and Dataset for Recognizing Emotional Responses in Children* (2016, November) http://multicomp.cs.cmu.edu/wp-content/uploads/2017/09/2016_ICMI_Nojavanasghari_Emoreact.pdf

[8] Najibi, A. *Racial Discrimination in Face Recognition Technology.* (2020, October 24) https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/

[9] Ciftci, U.A, Yuksek, G., & Demir, I. *My Face My Choice: Privacy Enhancing Deepfakes for Social Media Anonymization* (2020, November 2) *arXiv:2211.01361* https://arxiv.org/abs/2211.01361