Group 10 Final Report

Max Lisaius Bo Sullivan

Abstract

For this project we selected Task 1; audio speaker verification and Task 4; detection of GAN generated faces. We created and trained convolutional neural networks for both of these tasks to the greatest extent we could. For task 1 we selected a modified AlexNet [1] to train on slices of audio that we converted into spectrograms. This led to a resulting accuracy of approximately 94%, performing far beyond what random guessing or an inexpressive model could achieve. For task 4 we used a vanilla ResNet50 [2] and got an accuracy of 99.9%, far exceeding our expectations. This led us to begin investigating why the neural net was achieving such a high accuracy out-of-the-box. Overall we found with the right data augmentation, regularization, and model selection, convolutional neural networks provide accurate predictions for the tasks provided.

I. Task 1: Speaker Verification

A. Methods

- Data Preprocessing / Feature Extraction

  We start by creating text files of filename pairs, where approximately half of them are different audio clips of the same speaker and the other half is audio clips of two different speakers. We then pass these file pairs into our data loader so that each pair is one index. When a data point is requested, the audio files are loaded in and clipped down to a size set in the hyperparameters (default 25000).
  At first we always set this clip to the center of the full audio, but after we observed a large over fitting problem on the training data set, we now take this length from a random part of the audio clip. We then take our pair of 1-D waveforms and convert them to audio spectrograms with torchaudio's Spectrogram() transform.
  At this point we should have a pair of 2-D audio spectrograms that are the same size. We take these two images, and stack them depthwise such that the pair of spectrograms each inhabit their own channel.
- Models Developed

  The final model we went with was a modified Alexnet [1]. We found that overall when no pre-trained models are allowed, Alexnet [1] served as a good network to learn our channel layers and train quickly in a limited time. Once we added the random shifting to the dataloader it also did a good job generalizing.
  Another model we looked at and tried out was a Siamese network with a contrastive loss function. This network would take each picture and pass it though the net, embedding it into N dimensional feature space. Trained on a pairwise distance based contrastive loss, we never managed to get any traction, and the network would output mostly noise.
  We also looked at models like ResNet [2] and VGG, but did not have as much luck as we did with AlexNet [1].
- Training and Tuning

  Our model yielded mild results upon our first successful run; the modified AlexNet [1] got 72% validation accuracy and a test loss of .261. When we looked at ways of improving these numbers, we tried tuning with different optimizers, minibatch sizes, and learning rates with marginal improvement. We decided to increase our training set count by generating more pairings and then make use of hard example mining by changing the 50/50 split in the training set to be a 62.26/37.26 split for unalike and like pairings. We found that this improved metrics significantly and with some minor tuning.
- Baselines
  For our testing, we kept the dev data splits at 50/50. Initial under-performing resulting in poor metrics from 50/50 training set splits and a smaller training set sample size. We also tried training directly on waveform features, we were unable to create a model that was expressive enough to learn from this form of data.

B. Results

Our model produced the best validation accuracy with a learning rate of 0.00085, minibatch size of 16, input clipping, using an Adam optimizer while being trained with a modified AlexNet [1]. Our best results were 94.23% validation accuracy.

Results	
Model	**Accuracy**
Baseline / Guessing	50%
Modified Alexnet (Ours)	94%

C. Conclusions

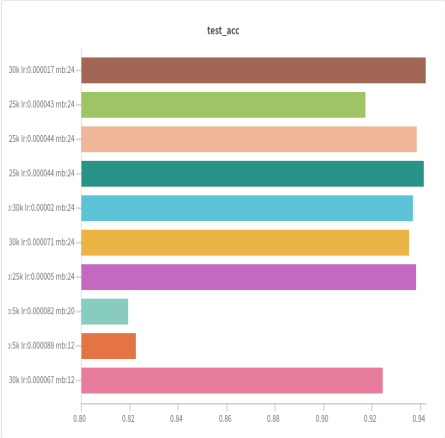
Through training task 1 we learned that a diverse dataset with sufficient size can lead to better generalization with our model. Through hard example mining and learning rate tuning we found the model generalized better.

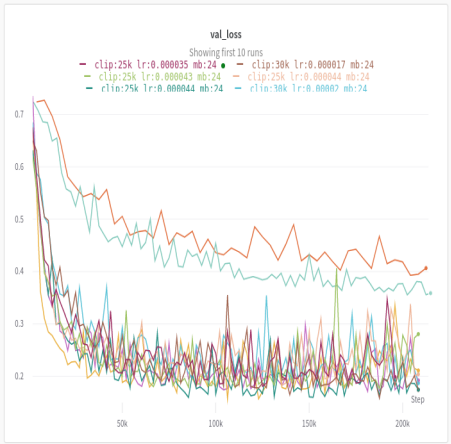
D. Contributions

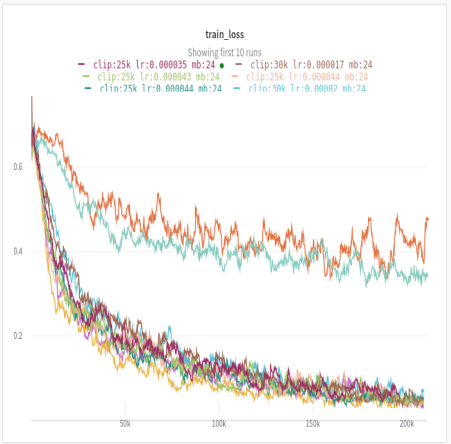
Bo worked on training models using CNNs, hyperparameter tuning of our modified AlexNet, hard example mining techniques and other pre-processing of training data. Max worked on model selection, regularization, pytorch-lightning implementation, data module for the spectrograms, and scripts to produce the txt files and for inference.

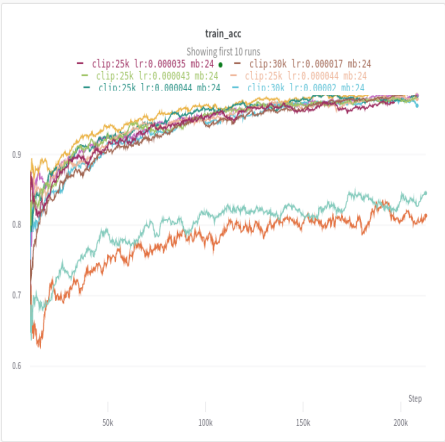
Task 1 Best Validation Accuracy
AlexNet 100 Epochs
LR 0.00085 MB 16

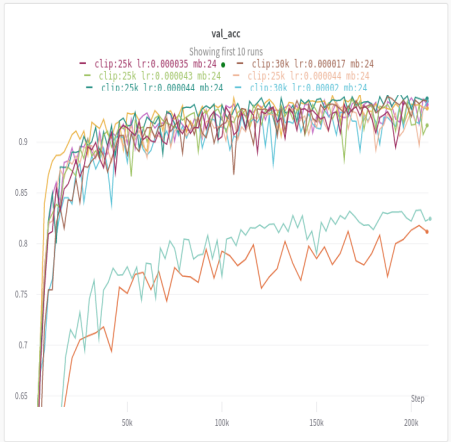
0.9423

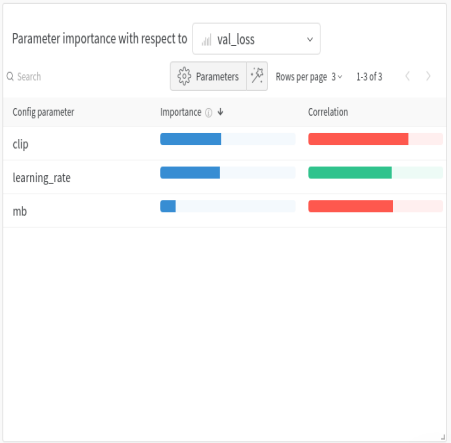


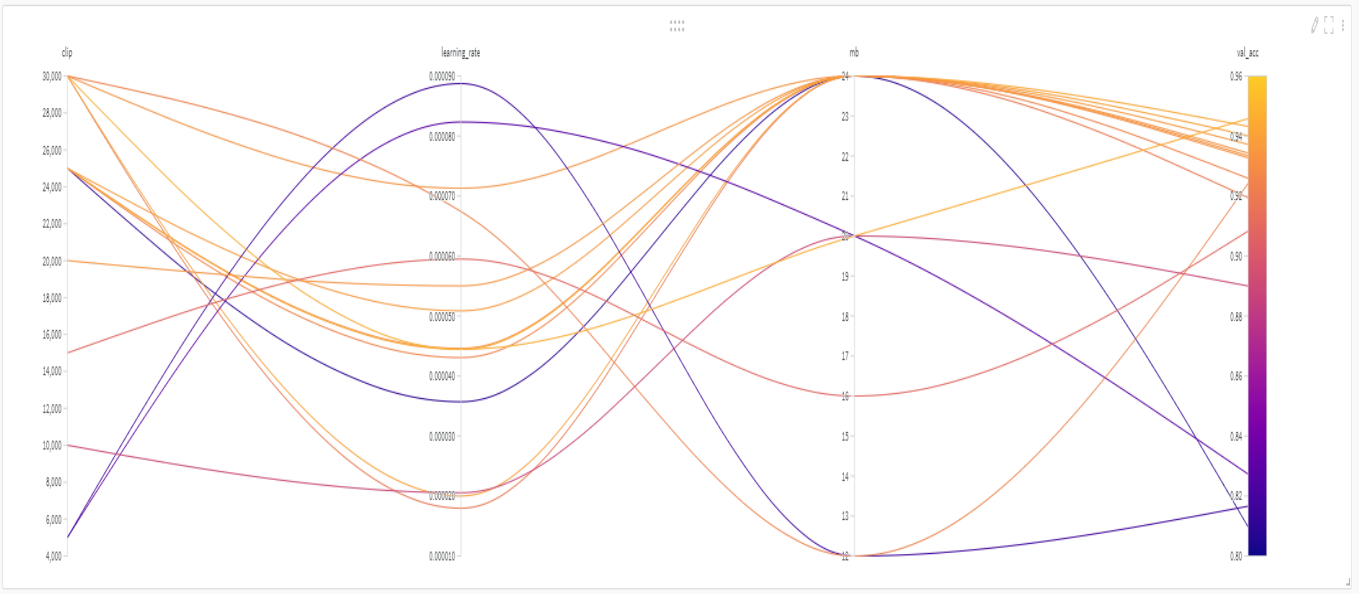












II. Task 4: Fake Image Detector

A. Methods

- Data Preprocessing / Feature Extraction

  For this task, we load each picture with PIL, convert it to a tensor, and return it with a label.
- Models Developed

  We tried multiple models like Alexnet [1] and VGG, but what we found worked best was ResNet50 [2]. While each epoch takes almost and hour, this model would achieve accuracies above 90% after the first epoch, with improvement falling off around 90-100 epochs.
- Training and Tuning

  Using ResNet50 [2] with this task, we found the best learning rate to be 0.00015 using the Adam optimizer. After about 10 epochs this would yield accuracy of over 99 percent. As mentioned before, training for this task took a very long time, the final model we are using for inference took over 48 hours to train. Below we can see some simple hyperparameter sweeps.
- Baselines

  In our initial experiments with inexpressive models, we observed noisy accuracy oscillating between 45-55%. Originally we thought that this initial uptick to 55% was training occurring, however these turned out to be random guesses. After looking at the dev data provided we noticed it is made up of 6923 real and 8401 fake images, meaning the dev split was 45.1% real and 54.9% fake.

B. Results

As we can observe in the figure on the next page, the model trained on 100 epochs performed the best. Additionally we can also see the diminishing returns from training past this point.

Results	
Model	**Accuracy**
Baseline / Guessing	54.9%
ResNet50 (Ours)	99.92%

In order to percieve how our model is performing so well we produced visualizations of the saliency or gradients of the neural network. While we cannot be certain, we can make some reasonable predictions with these provided maps even if the differences are small. We can observe the the fake map's activations mostly on the faces around the eyes and mouth. The real maps showing maps more uneven and dispersed. From this, we can gather that the model is picking up something about the facial features or symmetry of them. In this case the neural net may be recognizing the artifacts or in-painting imposed by the generator of the original network that generated these images, suggesting that this model is modern compared to the discriminator used by the GAN to produced these images. Another possibility is the data collection process needs to be reviewed, for diverse GAN imagery, face-centering, and head pose, among others.

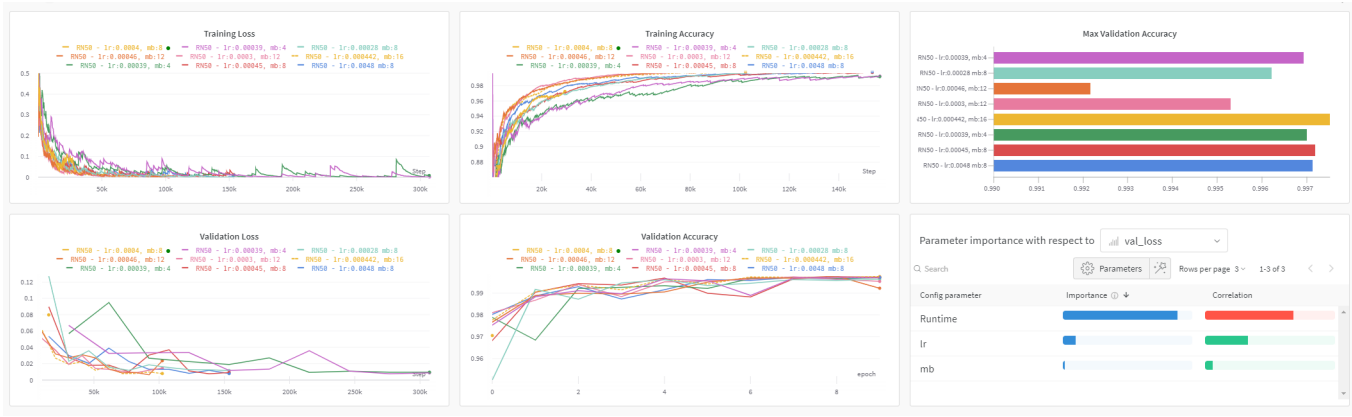
We also wanted to investigate the few remaining pictures that the model guessed incorrectly, as these might provide insight into how the model is making these approximations. We noticed a majority of the pictures are real images labeled as fake, and the subjects are primarily women. This suggests that the data may contain bias in its representation of women or that images that were miss-classified may have some post-processing applied to them. Another possibility is that these images were taken by professionals with manual focus lenses creating a blurry background similar to ones produced in GAN generated faces. This in turn may have deceived the model and may lead us to ask if some people have faces that may look more generated to one of these models than others? There is also the ethical consideration if these people may face disproportionate hardship in the creation of social media accounts or general digital identity verification in the future.

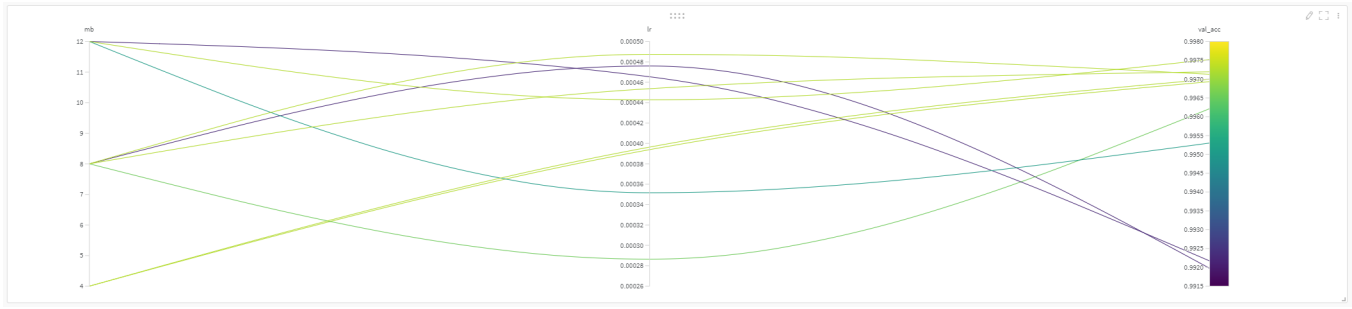
C. Conclusions

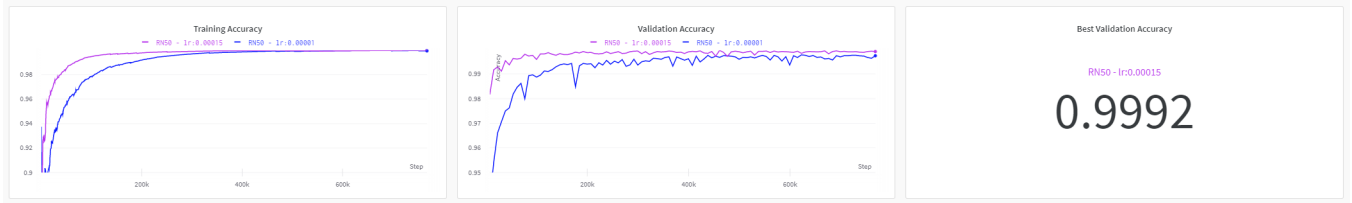
In this task we saw the power and efficacy of ResNet50 [2] in identifying faces produced by GAN networks. We found the best hyperparameters to be lr of 0.00015 and mb of 16, trained for as long as possible. Without knowing the origin of this data we cannot anticipate a comparison to state-of-the art, but we are confident in our findings and model given the time and compute restrictions.

D. Contributions

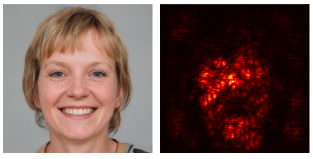
Max worked primarily on Task 4 while Bo helped validate results via testing.



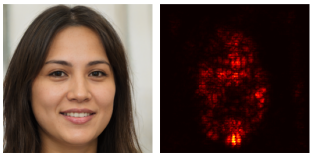




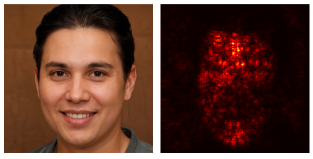
Fake Image and Saliency, Confidence: 1.0



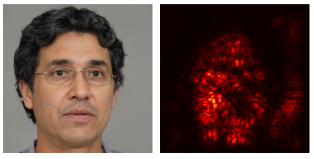
Fake Image and Saliency, Confidence: 1.0



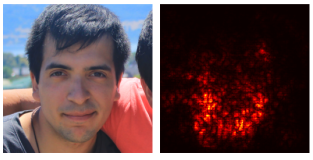
Fake Image and Saliency, Confidence: 1.0



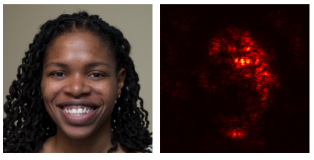
Fake Image and Saliency, Confidence: 1.0



Real Image and Saliency, Confidence: 1.0

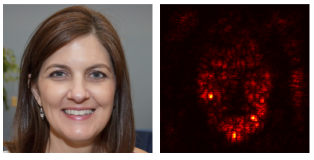


Real Image and Saliency, Confidence: 1.0

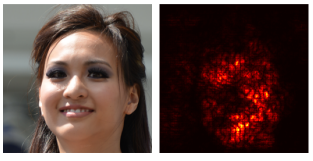


Real Image and Saliency, Confidence: 1.0

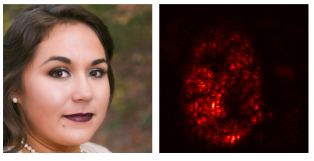
Real Image and Saliency, Confidence: 1.0



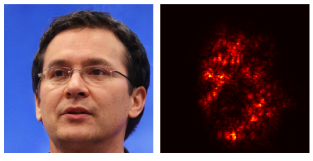
Real Image Labeled as Fake, Confidence: 0.6888635



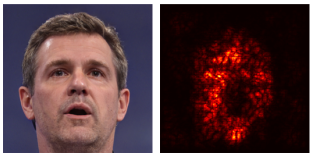
Real Image Labeled as Fake, Confidence: 0.99601185



Fake Image Labeled as Real, Confidence: 0.79226714



Fake Image Labeled as Real, Confidence: 0.504029



REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.