



CMPM17: Final Project



By Liam Youm & Kevin Nguyen
Team Name: Thing 1 & Thing 2



Agenda

- Problem/Goal
- Model Description
- Dataset Processing & Analysis
- Model Results
- Reflection & Challenges

Reminders

- Take notes on questions you want to ask by the end of the presentation!
- This format is a bit on the nose...

Problem/Goal

Create a model to distinguish between **6 facial expressions**:

- Angry
- Fear
- Happy
- Neutral
- Sad
- Surprise



Model Description

Model Description

CNN architecture

- 4 convolutional layers
- 3 linear layers
- 3x3 kernels & 1 padding on all convolutional layers
- 32 batch size
- 30 epochs
- Batch normalization after every layer
- ReLU activation function & Adam optimizer

Model Refinements/Iterations

Adding layers 3 and 4 (64 and 128 filters respectively)

- Grants complexity

10 -> 30 epochs

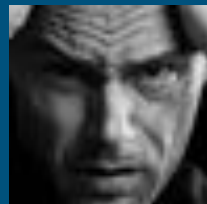
- Enough epochs for noticeable improvements; flattens afterwards

Dataset Processing & Analysis

About the Dataset (FER-2013)

Pretty organized already:

- 48x48 grayscale faces; already centered
- **Training Set:** 29,000 images
- **Test Set:** 3,600 images



Summary

► 📁 35.9k files

We did a few things after viewing...



1. Disgust & Anger Folders

We **merged** them together!

Why:

- Faces are similar
- Disgust folder has little samples
- CNNs online also struggled with this problem!

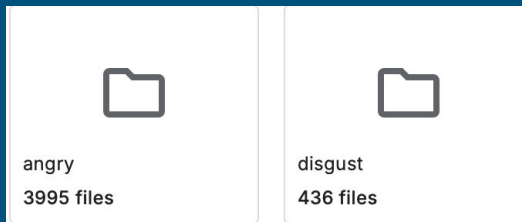
Disgust folder face that looks angry...



...and an angry folder face that looks disgusted???



train folder ->

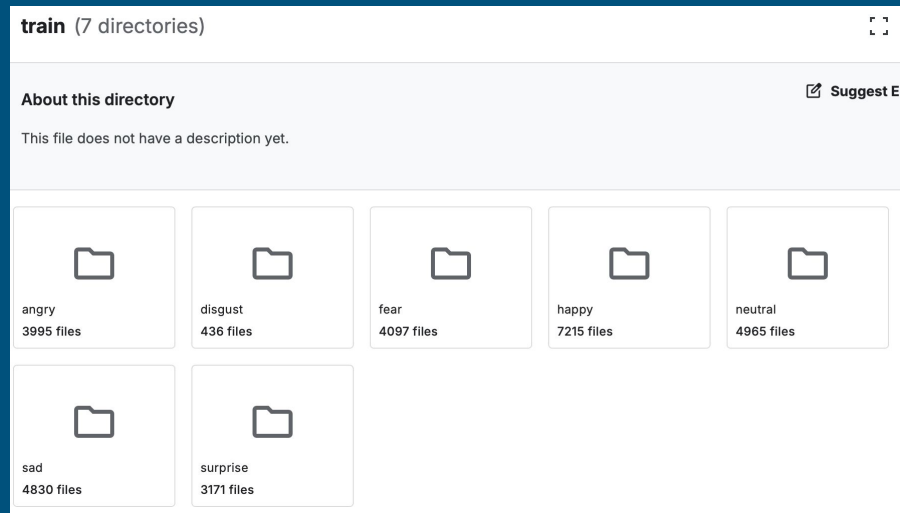


2. Happy Folder

We **cut** about 2000-3000 happy images

Why:

- Too many images
- May cause bias or make model worse if not cut
- Balances training



3. Image Channels

We used `v2.Grayscale(1)` on our tensors, even though it's already grayscale

Why:

- Tensor shape had 3 channels, but grayscale should only 1
- Properly makes them 1

```
return tf.nn.conv2d(
RuntimeError: Given groups=1, weight of size [16, 1, 3, 3], expected input[32, 3, 128, 128] to have 1 channels, but got 3 channels instead
stacktrace
```

Data Augmentations Used

```
train_transforms = v2.Compose(  
    v2.Resize((128, 128)),  
    v2.RandomHorizontalFlip(),  
    v2.RandomRotation(30),  
    v2.Grayscale(1),  
    #v2.Lambda(add_noise),  
    v2.ToTensor(),  
    v2.Normalize([0.5], [0.5])  
])
```

Mainly just rotations & flips

Facial expressions require a lot of information to predict, even from humans, so not many augmentations are needed!

Model Results

The Numbers

Test accuracy: ~60%

Average train accuracy: 66.9%

Average Loss: 1.09

Goals

Bare min accuracy: 25%

Initial target accuracy: 40%

Final target accuracy: 60%



(subtle foreshadowing)

Thoughts

Very good! Reached & surpassed expectations (60% target accuracy)

Notes: Possible overfitting, unstable learning



(subtle foreshadowing)

Demo Time!

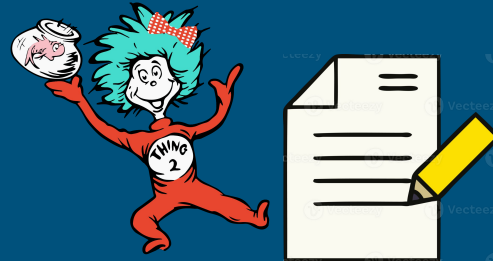
Reflection

What Went Well

Work ethic, punctuality, communication & feedback

Good system!

- Liam focused more often on advanced coding
- Kevin focused more often on code structure & documents



Challenges - What We Encountered

Rocky start - uneven work distribution

Kubernetes...



Liam Today at 7:08 PM

Kevin

it drives me crazy seriously

Challenges - How We Addressed Them

Feedback & self-reflection

Office hours, Discord server

Getting gooder





What We Would've Done Differently

Still try to even out work distribution

Keep up work ethic

Otherwise, not too much!

RECAP

- CNN to predict facial expressions
- Added layers & epochs to improve model
- Evened out classes for better training
- Model reached desired results
- Overall experience went well save a few difficulties

Thanks for listening!

Questions, comments, concerns?