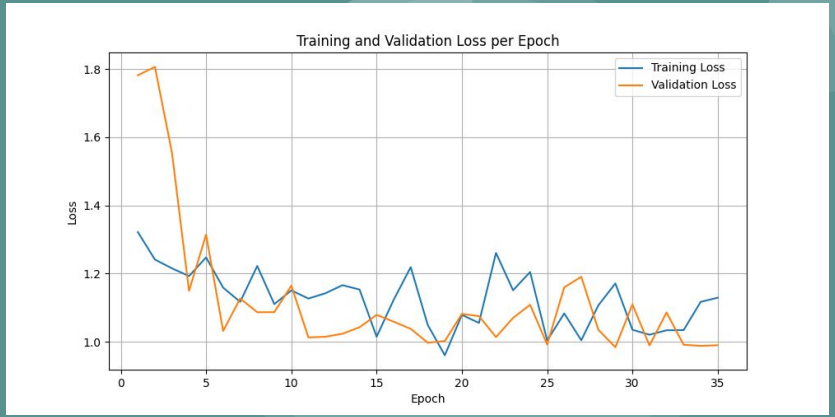
Small Project Presentation

Muhd Hafiz bin Abdul Halim (2302700)
Muhammad Azreen Bin Muhamamad (2200581)

CNN

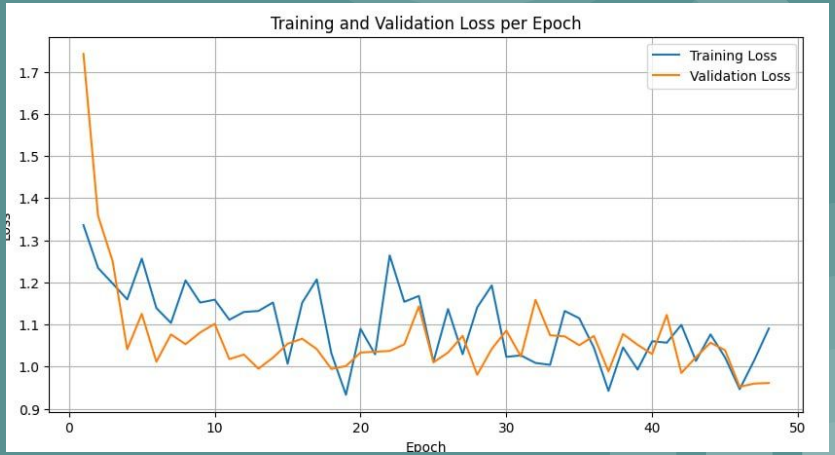
Changes (62%):

- Audio Augmentations
- NN Dropout set to 0.5
- Kernel Size [3, 3] because more efficient than [5, 5]



Changes (62.04%):

- Audio Augmentations
- Max Length = 8
- Learning Rate = 1.5e-4



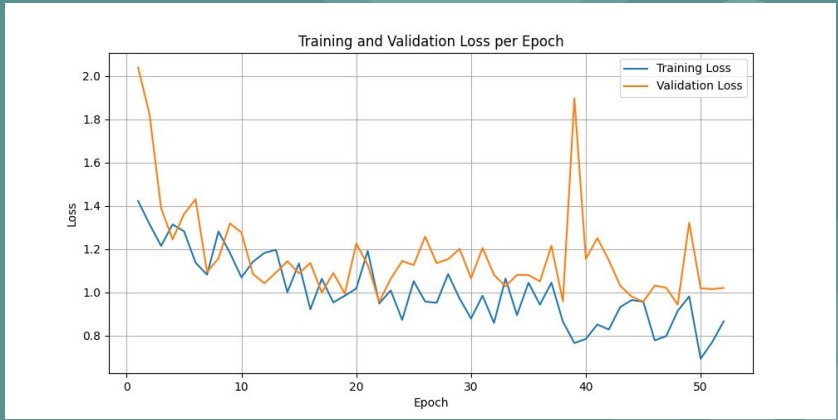
CNN

Changes (48%):

- Removed final CNN layer (in=256, out=512)
- Audio Augmentations
- Max Length = 8
- Learning Rate = 2e-4

Changes (61.5%):

- Added an additional CNN layer (in=512, out=1024)
- Audio Augmentations
- Max Length = 8
- Learning Rate = 2e-4



Pretrained (Wav2Vec2)

Why we choose Pretrained Wav2Vec2 Facebook?

- Wav2Vec2 is a self-supervised model that converts raw audio waveforms into meaningful speech representations for tasks like speech recognition.
- They provide a fair baseline for us to play without over relying on other tuned models.
- More room to fine tuned the process and experiment various approach

Our Model of choice

Bad Model E.g A IEMOCAP finetuned model online

Fine Tuning and Optimizer (Wav2Vec2)

- Preprocessing a dataset with a feature extractor before training is essential for several reasons, especially when working with audio, image, or text data.
- Models typically perform better when trained on features that highlight relevant patterns.
- You can save and load it for training future runs

```python
# Load the Wav2Vec2 feature extractor and model
feature_extractor = Wav2Vec2FeatureExtractor.from_pretrained("facebook/wav2vec2-base-960h")
model = Wav2Vec2ForSequenceClassification.from_pretrained(
    "facebook/wav2vec2-base-960h",
    num_labels=4  # Adjust this to match the number of emotion classes in your dataset
)

# Preprocess the dataset
def preprocess_function(examples):
    # Ensure that the inputs are numpy arrays and float32 type
    audio_inputs = [speech.astype(np.float32) for speech in examples["speech"]]
    # Use the feature extractor to process the audio inputs
    inputs = feature_extractor(audio_inputs, sampling_rate=16000, return_tensors="pt", padding=True)
    inputs["label"] = examples["label"]  # Add labels to the inputs
    return inputs
```

```python
# Load or process the dataset
LOAD = True

if LOAD:
    from datasets import load_from_disk
    emotion_dataset = load_from_disk('/kaggle/input/small-project-dataset/dataset/processed_dataset') #Change to Collab/Kaggle Directory
    print("Processed dataset loaded")
else:
    emotion_dataset = emotion_dataset.map(preprocess_function, batched=True)
    emotion_dataset.save_to_disk('./processed_dataset') #Change to Collab/Kaggle Directory

feature_extractor.save_pretrained("./wav2vec2-test") #Change to Collab/Kaggle Directory

def compute_metrics(pred):
    labels = pred.label_ids
    preds = np.argmax(pred.predictions, axis=1)
    accuracy = np.mean(preds == labels)
    return {"accuracy": accuracy}
```

Fine Tuning (Wav2Vec2) (68.81%)

Changes:

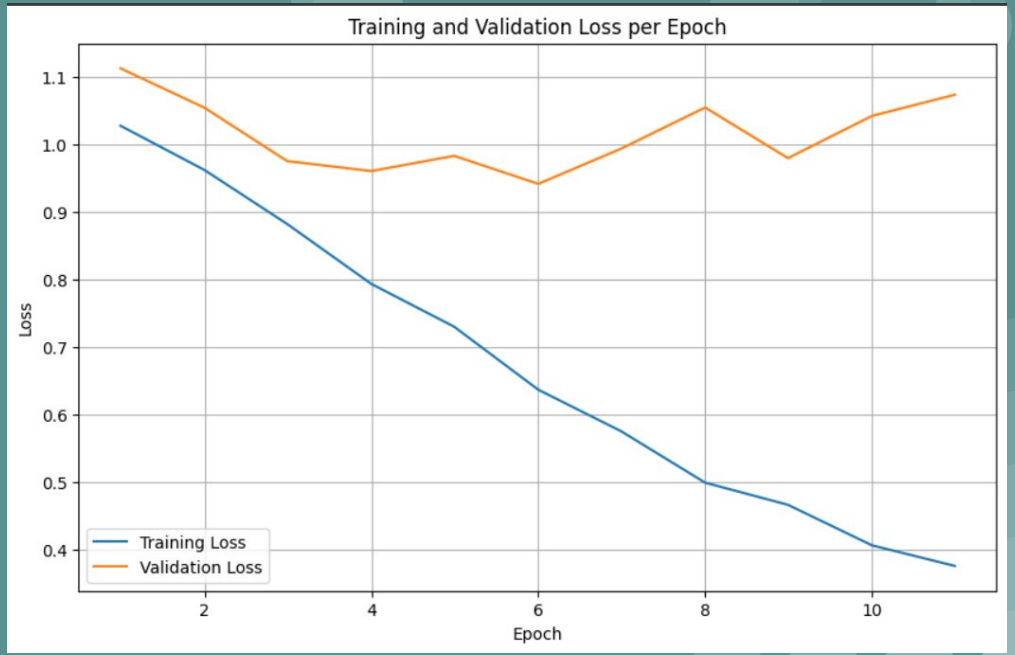
- max_len = 7
- epochs = 15
- weight_decay = 0.1
- learning_rate = 1e-5
- Used Adam Optimizer

Findings:

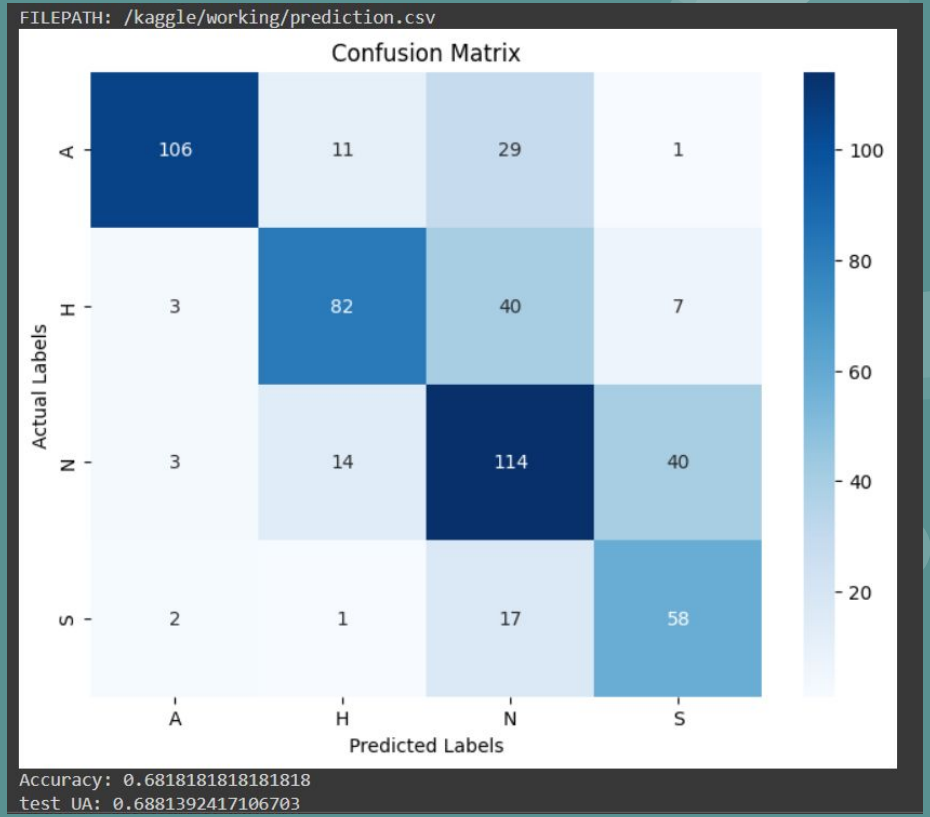
- Training Loss steadily increases, but Validation Loss begins fluctuating after epoch 4. Potential overfitting.
-

```
# Load the model
model = Wav2Vec2ForSequenceClassification.from_pretrained(model_save_path)

# Load the feature extractor
feature_extractor = Wav2Vec2FeatureExtractor.from_pretrained('facebook/wav2vec2-base-960h')
```



Confusion Matrix (Wav2Vec2) 68.81%



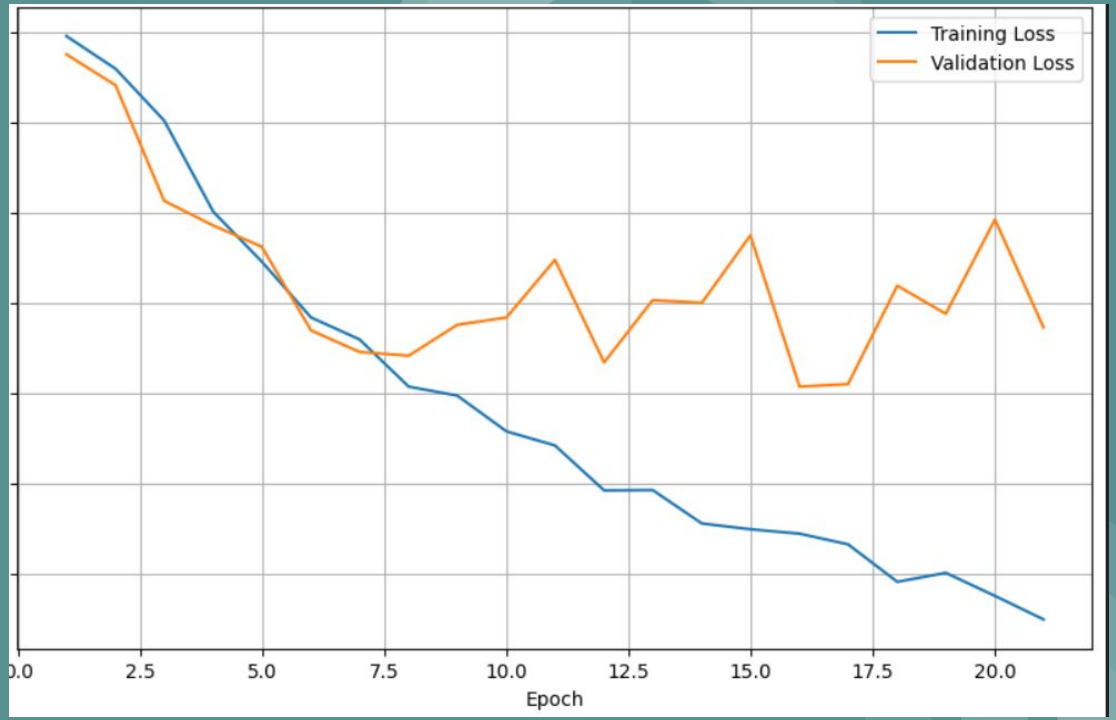
Fine Tuning (Wav2Vec2) (68.24%)

Changes:

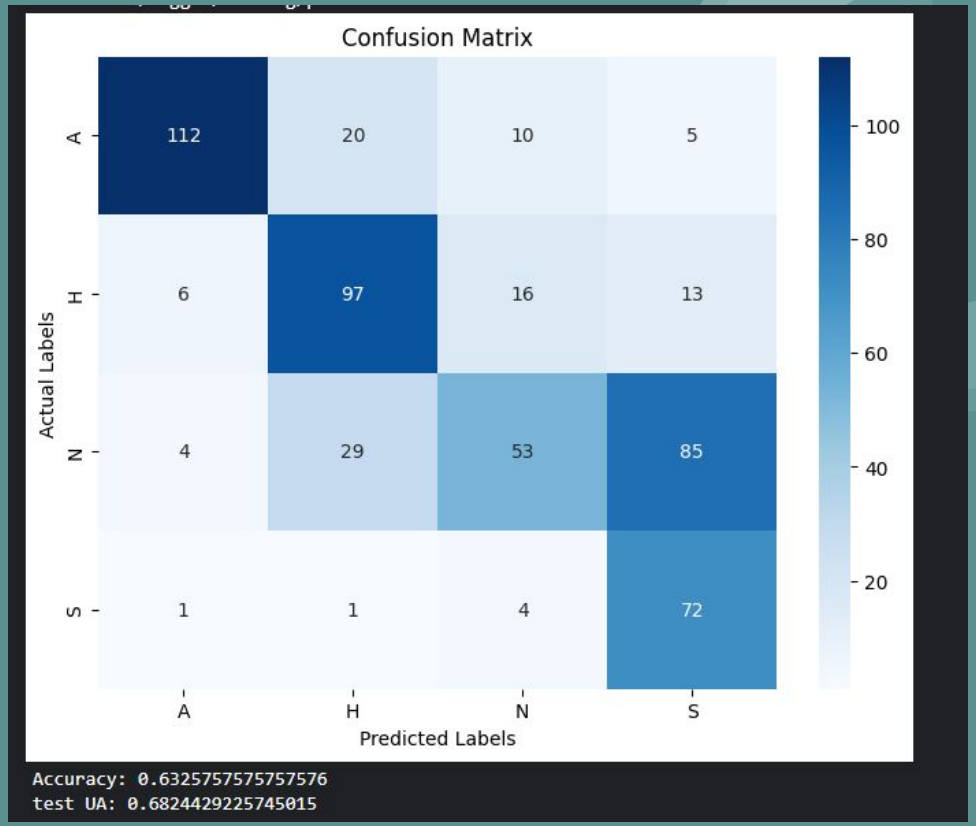
- max_len = 6
- Changed stopping to 10 epoch
- Increased the epoch run to 30

Findings:

- Consistent in where the model starts to generalize the data after a set amount of iterations.
- Adjusting the warm up step only delays or accelerate the time to it takes before generalization take place.

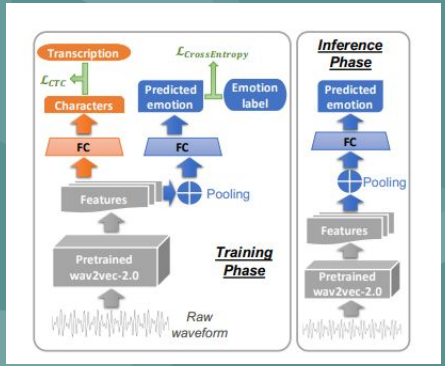


Confusion Matrix (Wav2Vec2) 68.24%



Other Attempts 2 Model Approach (Unsuccessful)

- Based my work on Baidu Research, USA
- https://www.isca-archive.org/interspeech_2021/cai21b_interspeech.pdf
- Have a second model (ASR) transcribe the audio to text to help the first model (SER) learn while training with Multi task learning layer to link both together




```
/content/dataset/train/Ses05F_impro02_M032.wav:I DON'T KNOW
/content/dataset/train/Ses05F_impro02_M033.wav:YERE IGUESS I DONOKNO WHAT THEY'RE OING TO SAY
/content/dataset/train/Ses05F_impro02_M034.wav:YE WE'LL CALL HIM WE'LL FIGER SOMETHING OUT I'M SORRY I'M SOW I'M SORRY YOU'RE RIGHT
/content/dataset/train/Ses05F_impro02_M035.wav:HE KNOW
/content/dataset/train/Ses05F_impro02_M036.wav:ALL THE TIME EVERY DAY EVERY DAY WHEN THEY HAVE EMALE OVER THERE IN SOME PLAKE THAT WRITE I CAN WILLLY SEND YOU PICTURES
/content/dataset/train/Ses05F_impro02_M041.wav:LOVEU
```

- Models like Wav2Vec2, Roberta and BERT have trouble telling voices with a heavy accent
- Encountered road block at the MTL layer portion causing a halt in this part of the project.
- Unable to to get both datatype to work together (Code attempt on colab)