ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
A- APPLIED SCIENCES AND ENGINEERING

Emotion Recognition through Speech Using Convolutional Neural Networks

Ilayda Su Irday[*]

[1]Engineering Department, Computer Engineering, Eskisehir Technical University, Eskisehir, Turkey.

ABSTRACT

In recent years, the intersection of technology and emotion recognition has gained significant attention. One fascinating area of research is the utilization of Convolutional Neural Networks (CNNs) to recognize emotions from speech signals. In this project, we explore the implementation of a CNN model for emotion recognition using the TESS Toronto emotional speech dataset.

Keywords: deep learning, convolutional neural networks, emotion recognition, pattern recognition

1. INTRODUCTION

1.1.Context

I'm on a journey to create an emotion classifier from audio and the TESS dataset is one of the 4 key datasets that I was lucky to stumble upon. What's interesting is that this dataset is female only and is of very high quality audio. Most of the other dataset is skewed towards male speakers and thus brings about a slightly imbalance representation. So because of that, this dataset would serve a very good training dataset for the emotion classifier in terms of generalization (not overfitting)

1.2. Content

There are a set of 200 target words were spoken in the carrier phrase "Say the word _' by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total. [10]

The dataset is organized such that each of the two female actor and their emotions are contain within its own folder. And within that, all 200 target words audio file can be found. The format of the audio file is a WAV format [10]

2. METHODS

2.1.Data Preprocessing:

The TESS dataset consists of audio recordings categorized into seven emotions: angry, disgust, fear, happy, neutral, surprise, and sad. The initial steps involve loading the data, extracting features from the audio files, and augmenting the dataset.

Librosa, a Python library for audio analysis, is employed to extract features such as mel spectrogram. Augmentation techniques include introducing noise, time stretching, shifting, and pitch variation.

```
path = "TESS Toronto emotional speech set data/TESS Toronto emotional speech set data/"
```

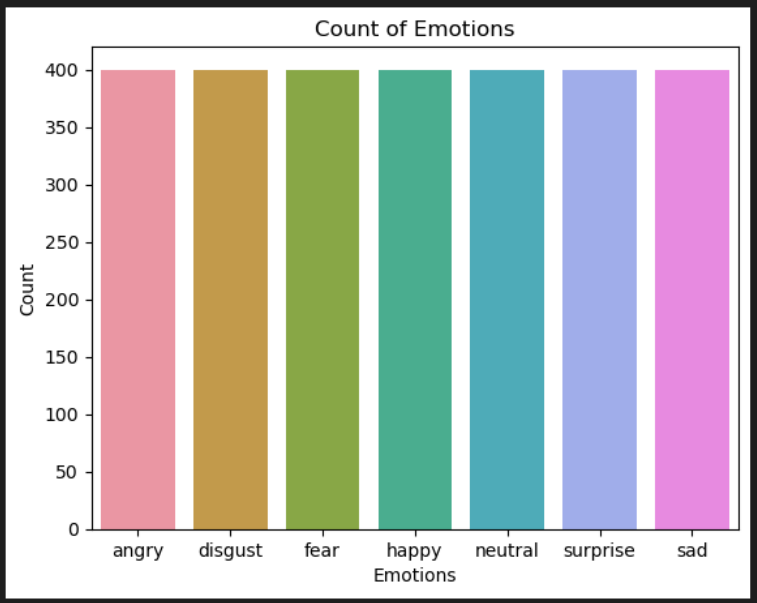
```
directory_list = os.listdir(path)
```

```
tess_directory_list = os.listdir(path)
file_emotion = []
file_path = []

for dir in tess_directory_list:
    directories = os.listdir(path+ dir)
    for file in directories:
        part = file.split('.')[0]
        part = part.split('_')[2]
        if part=='ps':
            file_emotion.append('surprise')
        else:
            file_emotion.append(part)
        file_path.append(path + dir + '/' + file)


emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
path_df = pd.DataFrame(file_path, columns=['Path'])
df = pd.concat([emotion_df, path_df], axis=1)
df.head()
```

	Emotions	Path
0	angry	TESS Toronto emotional speech set data/TESS To...
1	angry	TESS Toronto emotional speech set data/TESS To...
2	angry	TESS Toronto emotional speech set data/TESS To...
3	angry	TESS Toronto emotional speech set data/TESS To...
4	angry	TESS Toronto emotional speech set data/TESS To...

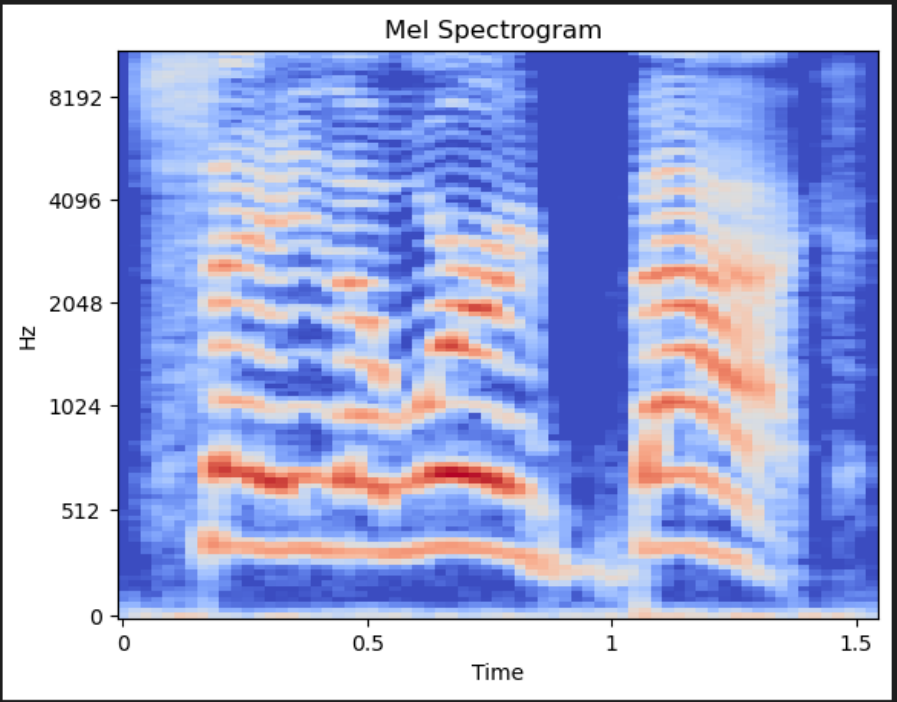
```
emotion_counts = df['Emotions'].value_counts()
sns.set_palette('spring')
plt.title('Count of Emotions')
sns.barplot(x=emotion_counts.index, y=emotion_counts.values)
plt.ylabel('Count')
plt.xlabel('Emotions')
```

```
Text(0.5, 0, 'Emotions')
```



```
spectrogram = librosa.feature.melspectrogram(y=data_array, sr=sample_rate, n_mels=128,fmax=5000)
log_spectrogram = librosa.power_to_db(spectrogram)
librosa.display.specshow(log_spectrogram, y_axis='mel', sr=sample_rate, x_axis='time')
plt.title('Mel Spectrogram ')
```

```
Text(0.5, 1.0, 'Mel Spectrogram ')
```




```
def noise(data):
    noise_amp = 0.045*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.shape[0])
    return data

def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data,rate=rate)

def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    return np.roll(data, shift_range)

def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(data, sr=sampling_rate, n_steps=pitch_factor)
```

2.2. Data Exploration:

After preprocessing, the dataset contains 5600 samples and each associated with a specific emotion label. The distribution of emotions is balanced, with 400 samples per emotion Category.

```
df['Emotions'].value_counts()
```

```
Emotions
angry       400
disgust     400
fear        400
happy       400
neutral     400
surprise    400
sad         400
Name: count, dtype: int64
```

2.3. Feature Scaling and Model Preparation:

To prepare the data for training, it is split into training and testing sets. Standard scaling is applied using sklearn's StandardScaler to normalize the features. The input shape of the data is then adjusted to fit the CNN model's requirements.

```
# splitting data
x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=42, shuffle=True)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
((4200, 162), (4200, 7), (1400, 162), (1400, 7))
```

```
# scaling our data with sklearn's Standard scaler
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
((4200, 162), (4200, 7), (1400, 162), (1400, 7))
```

```
# making our data compatible to model.
x_train = np.expand_dims(x_train, axis=2)
x_test = np.expand_dims(x_test, axis=2)
x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
((4200, 162, 1), (4200, 7), (1400, 162, 1), (1400, 7))
```

2.4 CNN Model Architecture:

The CNN model is designed with four convolutional layers, each followed by max-pooling to capture hierarchical features. Dropout layers are included to prevent overfitting, and dense layers provide the final classification. The model is compiled using the categorical crossentropy loss function and the Adam optimizer.[5]

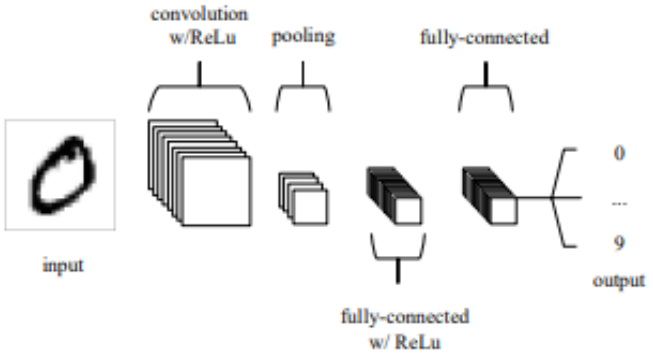
The basic functionality of the example CNN above can be broken down into four key areas.

2.4.1. As found in other forms of ANN, the input layer will hold the pixel values of the image.

2.4.2. The convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. The rectified linear unit (commonly shortened to ReLu) aims to apply Introduction to Convolutional Neural Networks 5 an 'elementwise' activation function such as sigmoid to the output of the activation produced by the previous layer.

2.4.3. The pooling layer will then simply perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation.

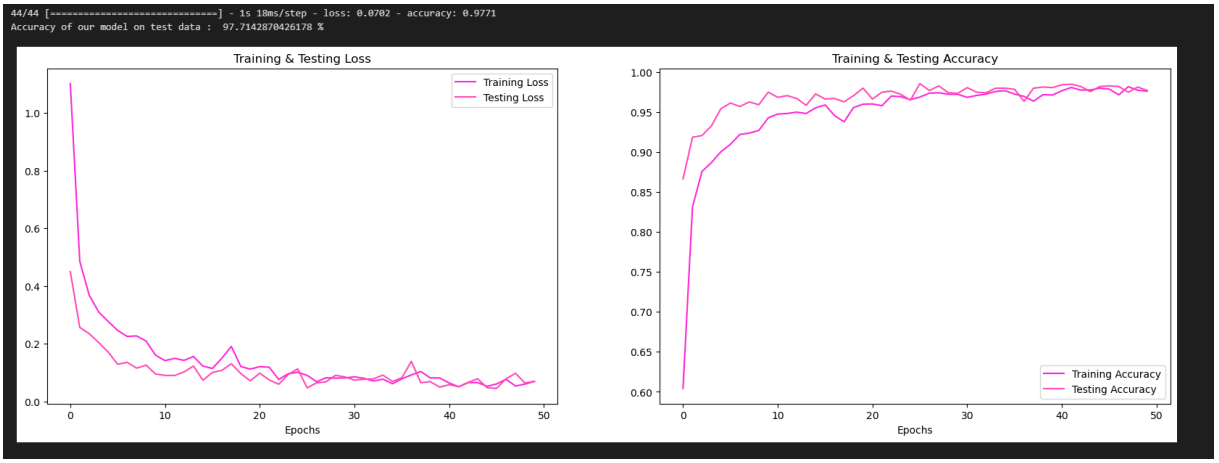
2.4.4. The fully-connected layers will then perform the same duties found in standard ANNs and attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance. [1]



Through this simple method of transformation, CNNs are able to transform the original input layer by layer using convolutional and downsampling techniques to produce class scores for classification and regression purposes.

2.5. Training and Evaluation:

The model is trained over 50 epochs, with a ReduceLROnPlateau[4] callback to adjust the learning rate dynamically. Training and testing accuracy and loss are monitored throughout the process.[6]



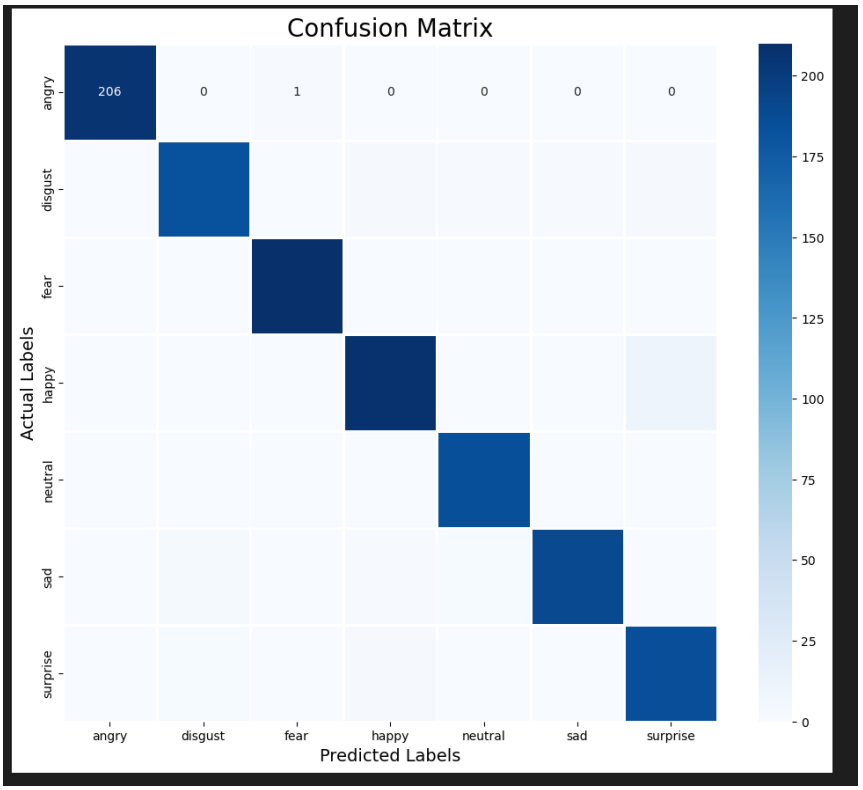
3. RESULTS

The model achieves impressive results on the testing set, with an accuracy of approximately 97.71%. This demonstrates the effectiveness of using CNNs for emotion recognition in speech.

When searching for a good classifier of a data set, and later, when tuning the selected classifier to achieve even better results, the data set is usually processed by applying cross-validation tests. The most general result of such a test is the averaged number of misclassified objects, which, however, provides no information on the recognition/classification of particular classes. And such information may be interesting, since the classification error is only very rarely uniformly distributed for all classes. Instead, some pairs of classes are relatively well distinguished by the classifier, while others are mostly confused. Easy identification of all such situations is one of the applications of the presented approach. Therefore it is useful to generate not only the averaged accuracy or averaged error of such test, but also what is called the averaged confusion matrix, i.e. the matrix that shows which classes have been confused with which during the test. Such matrix provides much more detailed information on the results of the test than the mere accuracy or error. It shows which classes were classified properly or almost properly and which where misclassified/confused with other classes and in what degree. Analysis of confusion matrices usually proves very useful.[8]

Accuracy is the simplest and most widely used metric to measure the performance of a classifier. However, accuracy is not always a good metric, especially when the data is imbalanced. The basic problem is that when the negative class is dominant, we can achieve high accuracy merely so long as we predict negative most of the time. As a consequence, even when a highly accurate classifier produces a positive prediction, it may still be that the negative class is (much) more likely. Such situations are characterized by precision of 50% or less. In many applications this is highly undesirable[9]

A precise relationship between precision, recall, and accuracy (that holds with equality) was discovered previously by Alvarez (2002), and our bounds could also be derived easily from Alvarez's equation. Interestingly, however, Alvarez did not interpret the consequences of his relation for the amount of data needed, which is our main contribution. Davis and Goadrich (2006) similarly showed that classifiers that dominate in the usual ROC space also dominate in precision-recall and vice-versa. Actually, results by Cortes and Mohri (2004) imply that precision-recall bounds are fundamentally different from AUC: for a fixed class imbalance and accuracy, various ranking functions may feature substantially different AUC scores, in contrast to precisionrecall bounds. [9]


Confusion Matrix
Actual Labels
Predicted Labels
angry
disgust
fear
happy
neutral
sad
surprise
206
0
1
0
0
0
0
200
175
150
125
100
75
50
25
0

```
df.head(10)
```

	Predicted Labels	Actual Labels
0	neutral	neutral
1	happy	happy
2	surprise	surprise
3	sad	sad
4	angry	angry
5	happy	happy
6	neutral	neutral
7	neutral	neutral
8	disgust	disgust
9	happy	happy

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       angry       1.00      1.00      1.00       207
     disgust       0.97      0.96      0.96       191
        fear       1.00      1.00      1.00       210
       happy       0.97      0.95      0.96       219
     neutral       0.98      1.00      0.99       185
         sad       0.99      0.96      0.98       197
    surprise       0.93      0.97      0.95       191

    accuracy                           0.98      1400
   macro avg       0.98      0.98      0.98      1400
weighted avg       0.98      0.98      0.98      1400
```

5. CONCLUSION

In conclusion, this article provides insights into the application of Convolutional Neural Networks for emotion recognition in speech. The TESS dataset, preprocessing techniques, and model architecture contribute to building a robust system. Emotion recognition through speech holds promise in various applications, from human-computer interaction to mental health monitoring. Further research and refinement of models could lead to even more accurate and reliable emotion recognition systems

CONFLICT OF INTEREST

The author(s) stated that there are no conflicts of interest regarding the publication of this article.

AI DECLARATION

Ai tools only used for linguistic check.

REFERENCES

[1] O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." *arXiv preprint arXiv:1511.08458* (2015).

[2]Abdel-Hamid, Ossama, et al. "Convolutional neural networks for speech recognition." *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014): 1533-1545.

[3] Huang, Jui-Ting, Jinyu Li, and Yifan Gong. "An analysis of convolutional neural networks for speech recognition." *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015.

[4] Al-Kababji, Ayman, Faycal Bensaali, and Sarada Prasad Dakua. "Scheduling techniques for liver segmentation: Reducelronplateau vs onecyclelr." *International Conference on Intelligent Systems and Pattern Recognition*. Cham: Springer International Publishing, 2022.

[5] Murphy, John. "An overview of convolutional neural network architectures for deep learning." *Microway Inc* (2016): 1-22.

[6]Hossin, Mohammad, and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." *International journal of data mining & knowledge management process* 5.2 (2015): 1.

[7] Boynukalın Z. Emotion analysis of Turkish texts by using machine learning methods. MSc, Middle East Technical University, Ankara, Turkey, 2012.

[8] Susmaga, Robert. "Confusion matrix visualization." *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM '04 Conference held in Zakopane, Poland, May 17–20, 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[9] Juba, Brendan, and Hai S. Le. "Precision-recall versus accuracy and the role of large data sets." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

[10] https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess