Horn Detection for Deaf Drivers

Alana Jocelyn Natania Massie 22/496239/PA/21331
Vian Sebastian Bromokusumo 22/496698/PA/21355

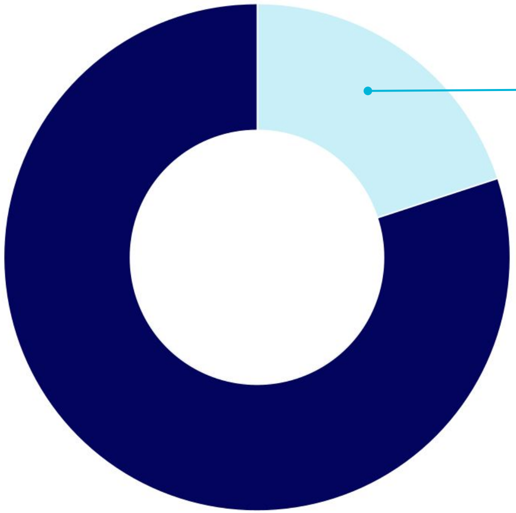
Pattern Recognition Course

Table of contents

01

Background

Background



20% of the population has some degree of hearing loss

1.16 billion of the 1.50 billion people who have hearing loss have mild severity

Around 70 million people globally have total hearing loss, and this is **projected to double by 2060**

83.3% of deaf drivers face **impaired awareness** during driving, leading to 60% experiencing accident

Underlying Problems for Deaf Drivers

High interest for IoT devices that increase perception and driving awareness

Hesitance to acquire drivers license due to fear of safety

Higher average travel distance due to slower routes and increased direction confusion

SDGs

3 GOOD HEALTH AND WELL-BEING

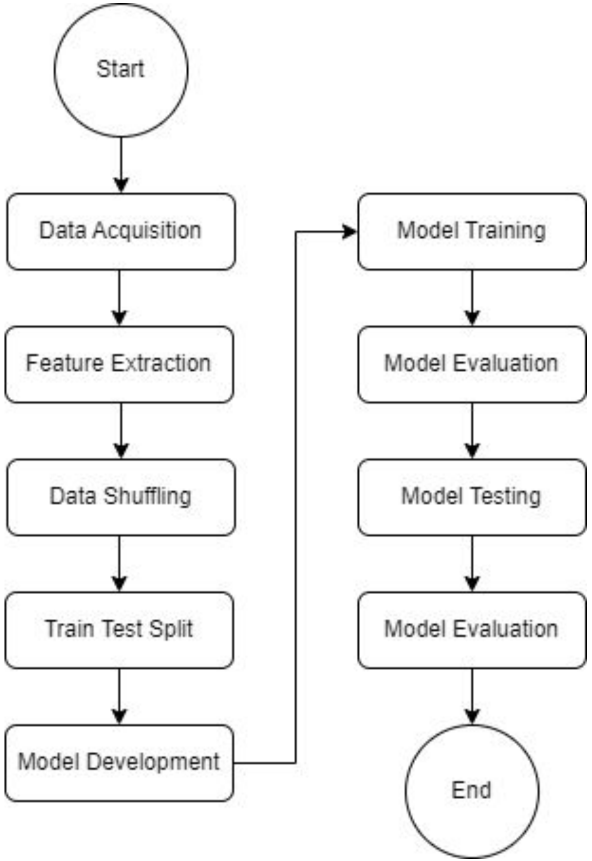
10 REDUCED INEQUALITIES

11 SUSTAINABLE CITIES AND COMMUNITIES

02

Method

Project Flowchart



Dataset

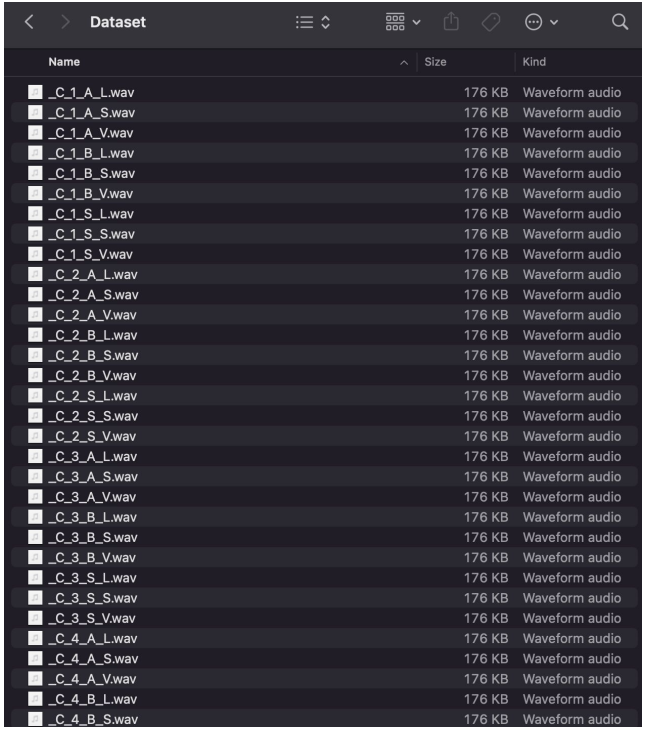
HornBase

A dataset comprising 1080 audio files of exactly one-second duration each

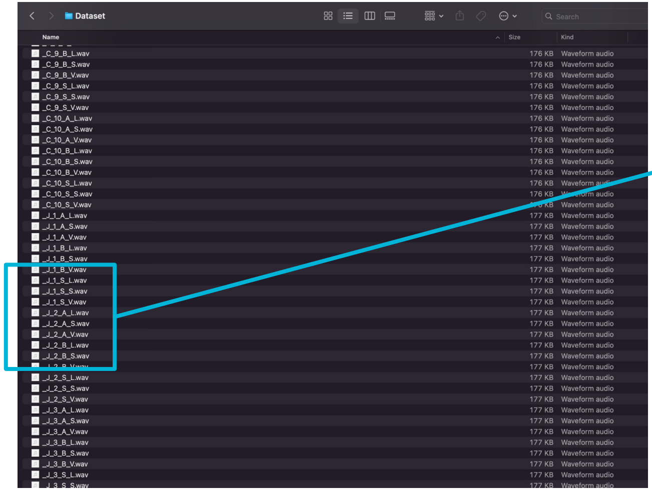
	Horn	Not Horn
Classes	– Short honk – Long honk – Intermittent sequence of three consecutive short honks	– Road noise – Music noise – Talking noise

For each possible audio segment, **three temporal windows are cut**, with the first containing the initial half of a horn, the second containing the entire horn, and the third containing the final half of a horn.

Contributors:
Cleyton Aparecido Dim, Nelson Cruz Sampaio Neto, Jefferson Magalhães de Morais



Dataset



Filename Structure

```
=HORN CLASS=

_C_9_B_L_.wav
||  |  |  ||
||  |  |  ||___ present only if is second half horn cut
||  |  |  |____ type of the horn. (S)hort, (L)long, (V)aried
||  |  |_______ position of the emitting car. (B)ack, (S)ide, (A)head
||  |__________ scenario identifier. Ranging from 1 to 10
||_____________ recording smartphone identifier. (C or J)
|______________ present only if is first half horn cut


=NOT HORN CLASS=

C_9_B_N3.wav
| |  |  |
| |  |  |_____ range from 1 to 9 to differentiate not-horn cuts
| |  |________ position of the emitting car. (B)ack, (S)ide, (A)head
| |___________ scenario identifier. Ranging from 1 to 10
|_____________ recording smartphone identifier. (C or J)
```

Dataset Preparation

```
for file in files:
if os.path.isdir(os.path.join(source_dir, file)):
continue

if file.endswith(('S.wav', 'S_.wav', 'V.wav', 'V_.wav','L.wav',
'L_.wav')):
target_dir = honks_dir
else:
target_dir = non_honks_dir

shutil.move(os.path.join(source_dir, file),
os.path.join(target_dir, file))

print('Files in Honks:', os.listdir(honks_dir))
print('Files in Non-Honks:', os.listdir(non_honks_dir))
```

The code snippet groups the dataset into the correct structure as shown in the figure

Folder Structure

```
Dataset/
     |
     |---Honks/
     |
     |---Non-Honks/
```

Feature Extraction

What?

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

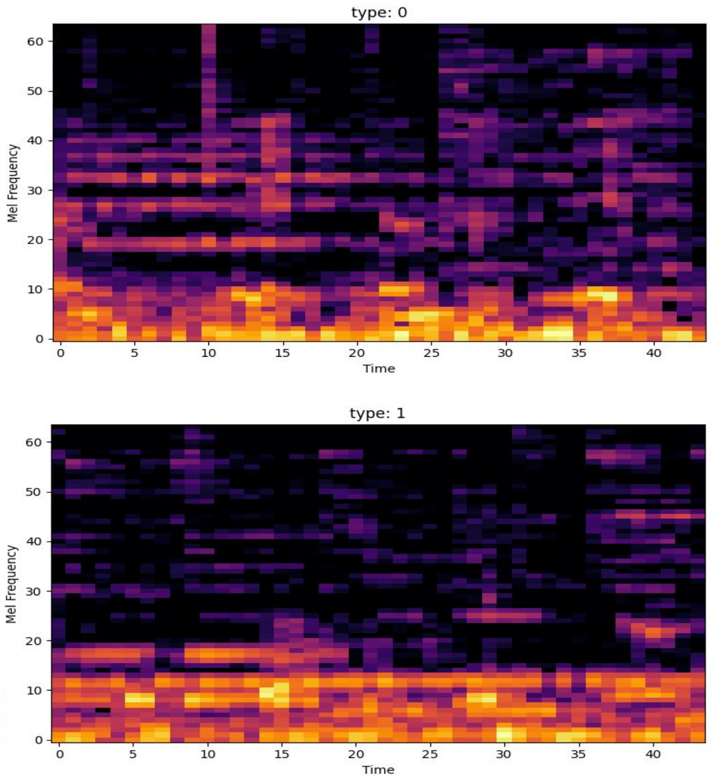
How?

The main feature we extracted from the audio files are the spectrograms, the sample of each class can be seen in the next slide.

The type 0 will be the Horns, and the type 1 will be the Non-Horns.

Feature Extraction

```
for genre, genre_number in genres.items():
    directory = os.path.join(base_dir, genre)
    genre_data = []
    for filename in os.listdir(directory):
        files = os.path.join(directory, filename)
        for index in range(14):
            y, sr = librosa.load(files, mono=True, duration=2)
            ps = librosa.feature.melspectrogram(y=y, sr=sr,
hop_length=512, n_fft=512, n_mels=64)
            ps = librosa.power_to_db(ps**2)
            genre_data.append({'spectrogram': ps, 'type':
genre_number})
    genre_df = pd.DataFrame(genre_data)
    dfs.append(genre_df)
```



Train Test Split

Shuffling

From the initial 15,120 spectrograms data, we first applied random data shuffling to mix the data with no particular data. This is to ensure the integrity of the data

Three-way split

Ratio of 90:10 (training - temp)
Ratio of 60:40 (eval - test)

Reshaping

The x values are stacked to create a 3-channel input, while the y data are encoded

```
shuffled_df = df.sample(frac = 1, random_state = 42)
shuffled_df

x_train, x_temp, y_train, y_temp = train_test_split(x, y,
test_size = 0.1, random_state = 123)
x_val, x_test, y_val, y_test = train_test_split(x_temp,
y_temp, test_size = 0.4, random_state = 123)

x_train = np.repeat(x_train, 3, axis=-1)
x_val = np.repeat(x_val, 3, axis=-1)
x_test = np.repeat(x_test, 3, axis=-1)

y_train = tf.keras.utils.to_categorical(y_train,
num_classes=2)
y_val = tf.keras.utils.to_categorical(y_val, num_classes=2)
y_test = tf.keras.utils.to_categorical(y_test, num_classes=2)
```

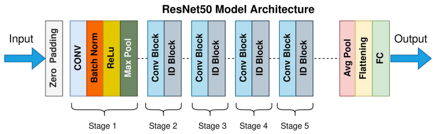
Model Development

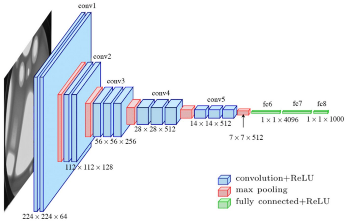
Extra Layers

Compilation

ResNet50

VGG16





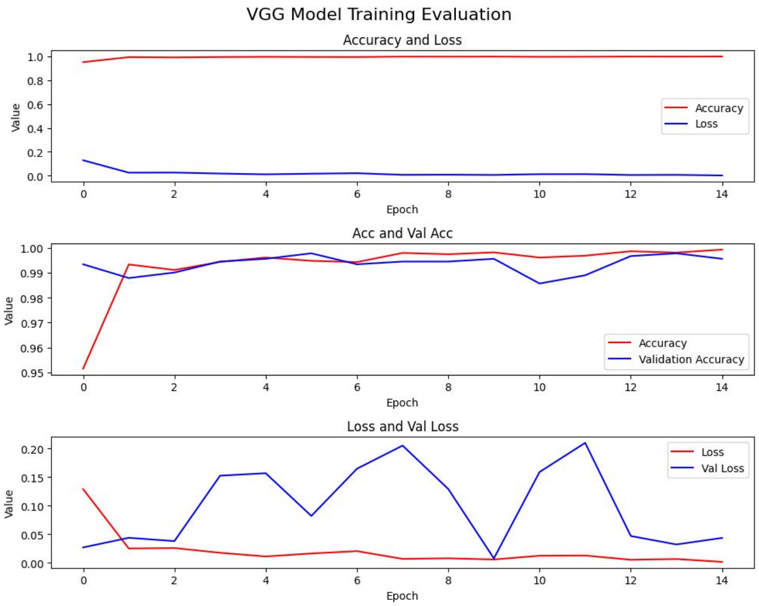
Model Development

VGG16	VGG-16 is a deep learning network consisting of 16 weight layers: 13 convolutional layers and 3 fully connected layers. The network uses small 3x3 filters in all convolutional layers and max-pooling layers to reduce dimensionality.
ResNet50	ResNet50 is a 50 layer deep learning network consisting of convolutional layers, identity mappings, and pooling layers. Key advantage of the ResNet network is the presence of residual blocks that effectively mitigates the vanishing gradient problem.
Extra Layers	**Batch Normalization, Dropout, and two Dense layers for dataset customization.**
Compilation	**Categorical Crossentropy, Adam optimizer, and accuracy metric.**

03

Results & Discussion

VGG16 Training Result



Highest recorded Accuracy: 0.9993
Highest recorded Val Accuracy: 0.9978

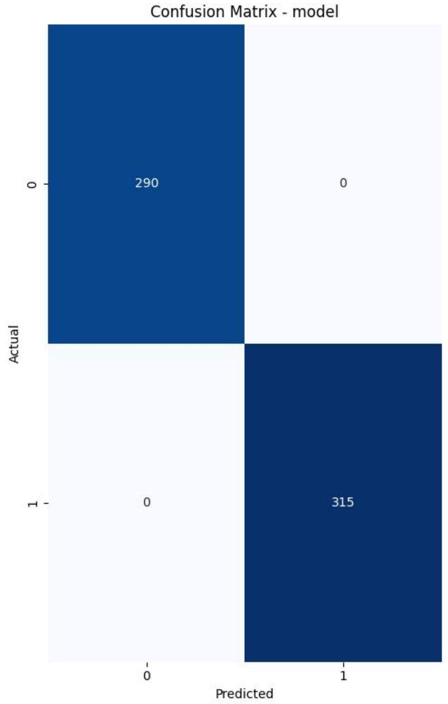
ResNet 50 Training Result

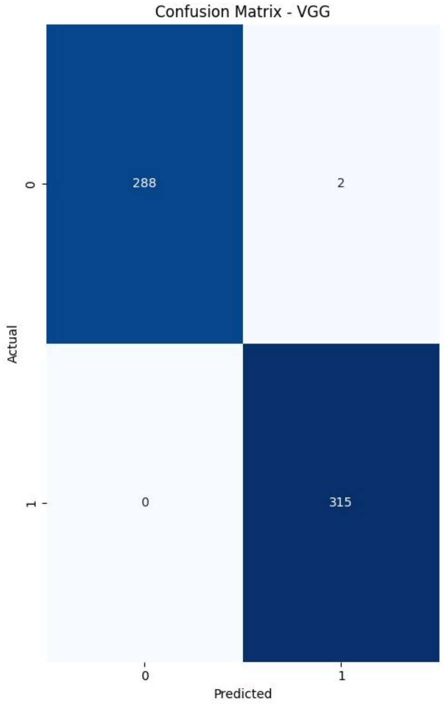
Highest recorded Accuracy: 0.9999
Highest recorded Val Accuracy: 1.000

Testing Results

	ResNet	VGG
Acc	1.0	0.9966
Prec	1.0	0.9967
Rec	1.0	0.9966
F1	1.0	0.9966

Highest recorded Accuracy: 0.9993
Highest recorded Val Accuracy: 0.9978





Discussion

	ResNet50	VGG16
Acc	1.0	0.9966
Prec	1.0	0.9967
Rec	1.0	0.9966
F1	1.0	0.9966

	ResNet50	VGG16
Size	98 MB	528 MB
Params	98.6 M	138.4 M
Trainable Params	4 MB	1 MB
Inference time	4.6 ms	4.2 ms

From this project, it is undisputed that the **ResNet50 model performed better,** with **less space complexity**, making it a better suit in the ideally proposed solution in deploying the model to a mobile environment. Although, it is worth mentioning that both these models are very sophisticated deep learning models, which might be an overkill.

04

Conclusion

Conclusion

Considering the accuracy, size, and many other factors, both ResNet50 and VGG16 output outstanding results in classifying and detecting horn sounds from other noises. ResNet50 performed better and is significantly more space-efficient, with a smaller model size and fewer parameters. Although ResNet50's inference time is marginally longer, its overall performance and lower space complexity make it better suited for deployment.

It can be concluded that this project effectively created a horn detection system with the aim to help deaf drivers in the driving environment.

Future Recommendations

Real-world testing

Have deaf drivers test the models to evaluate efficiency in practical scenarios and gather feedback for improvements.

Hybrid Architectures

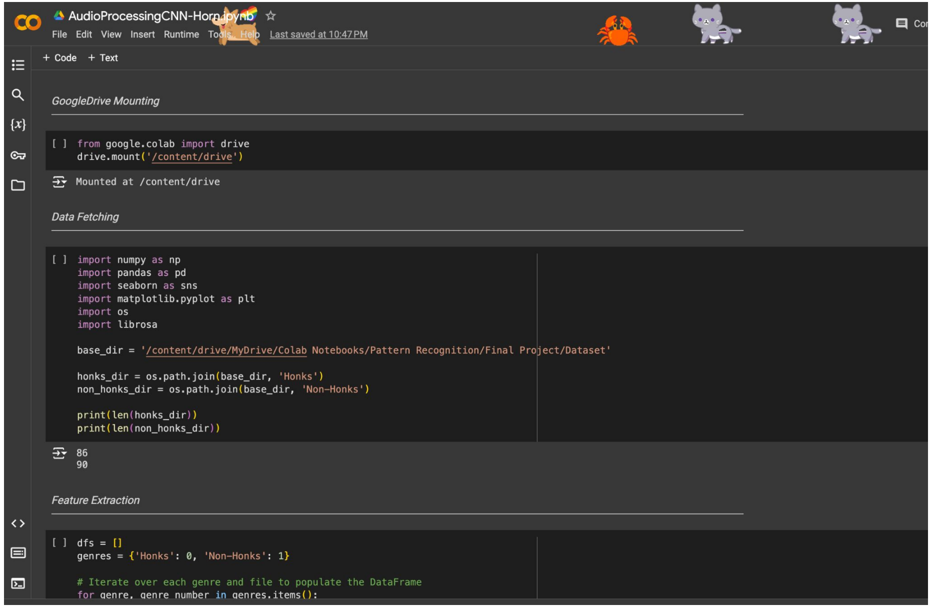
Explore combining VGG16 and ResNet50 features or using newer architectures like EfficientNet or Vision Transformers.

Dataset Augmentation

Find more extensive and diverse datasets and apply data augmentation techniques for better generalization.

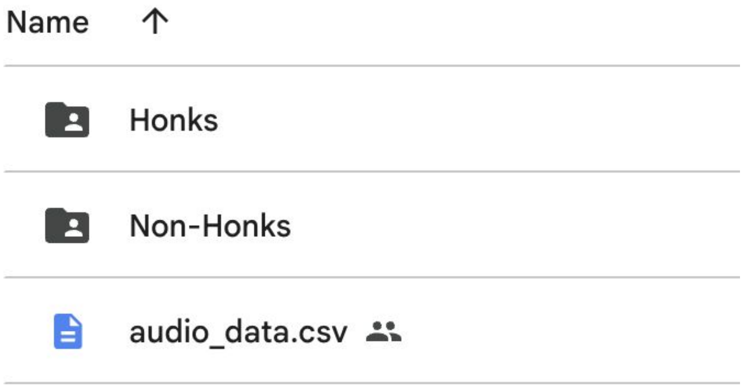
Project Link

https://colab.research.google.com/drive/1W_0KWlmaiSbr5_QPc3mA-vhig83r2YYF?usp=sharing



Dataset Link

https://drive.google.com/drive/folders/1c2L7efiMOzDqa98EVlNGwKuIZoAFUJP1?usp=sharing



References

1. Dim, Cleyton Aparecido; Neto, Nelson Cruz Sampaio; de Morais, Jefferson Magalhães (2024), "HornBase - A Car Horns Dataset", Mendeley Data, V2, doi: 10.17632/y5stjsnp8s.2
2. MathWorks. (n.d.). What is feature extraction? MathWorks. Retrieved June 11, 2024, from https://www.mathworks.com/discovery/feature-extraction.html
3. Mascarenhas, S., & Agarwal, M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON-2021) IEEE. https://doi.org/10.1109/CENTCON52345.2021.0017

Thank You

Any questions?