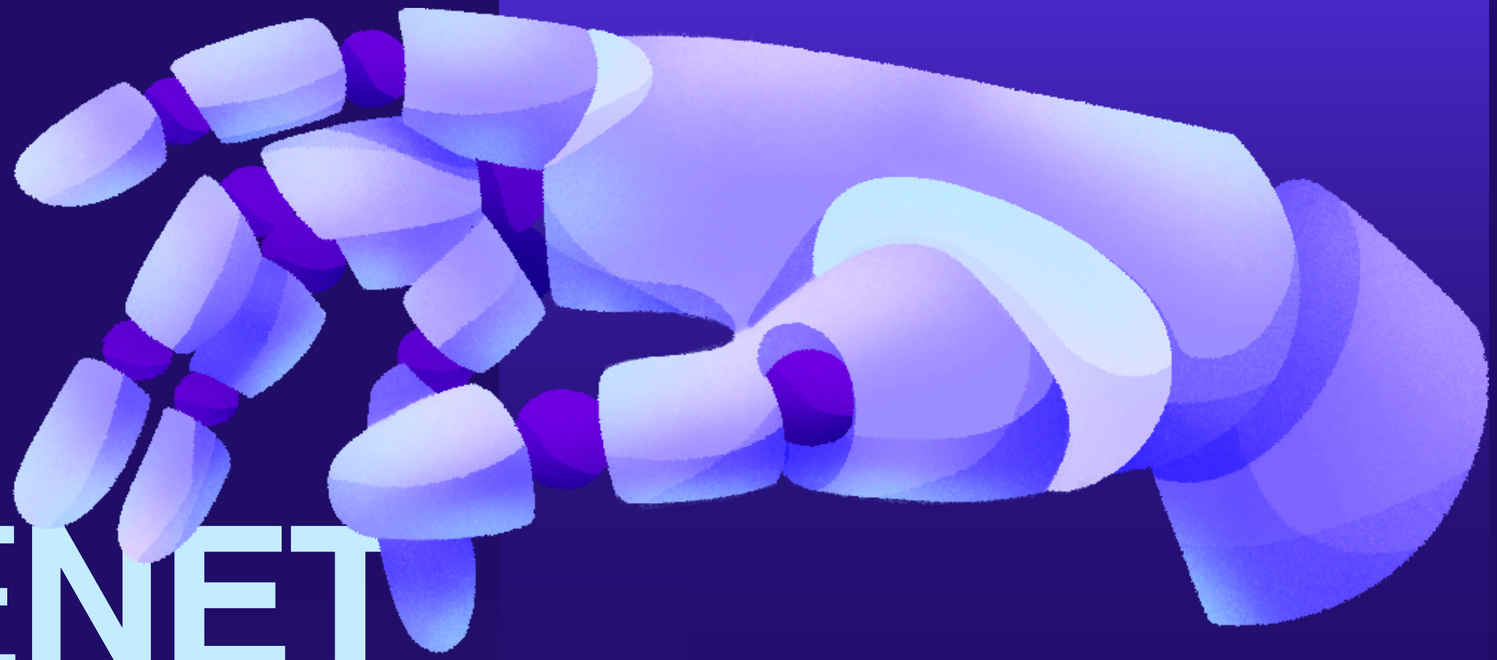


VEHICLENET

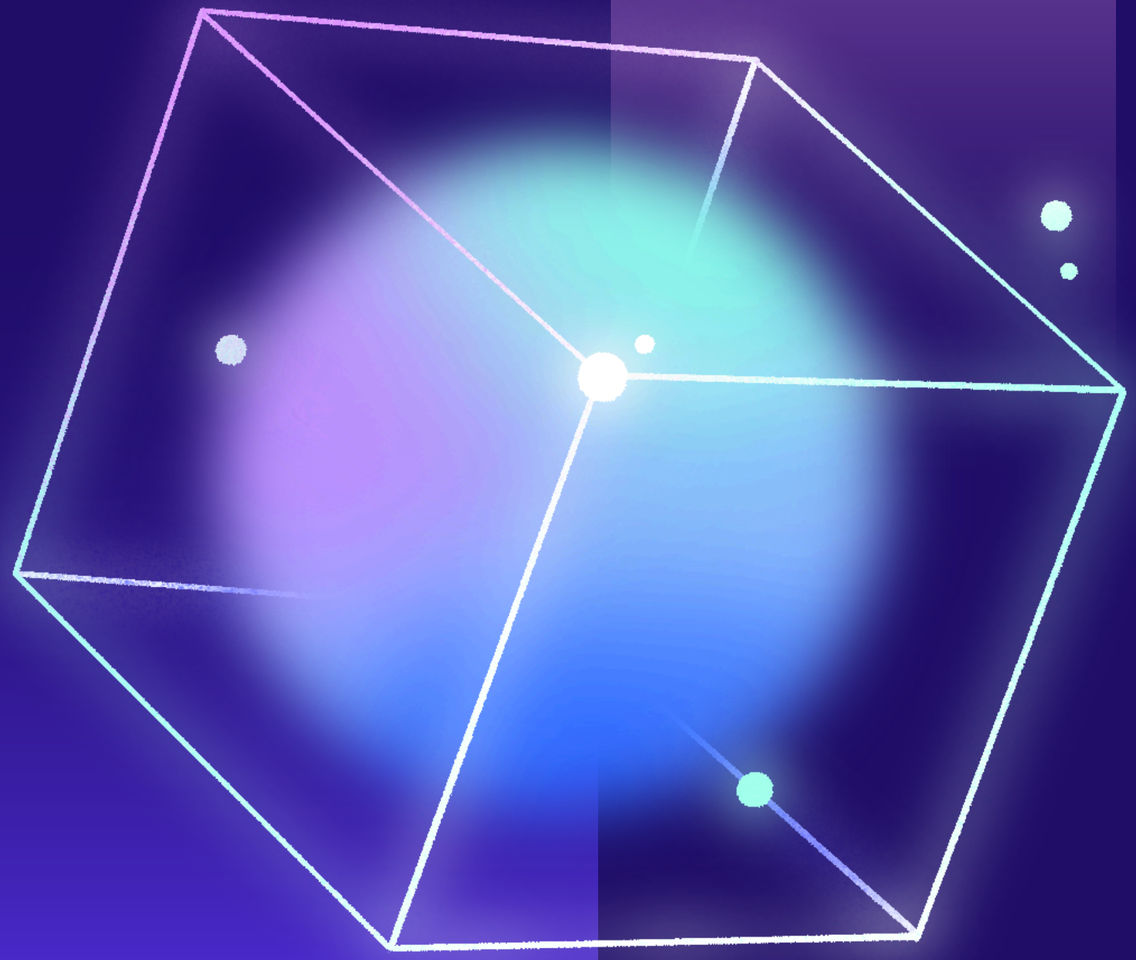
ENGINE TYPE CLASSIFICATION

APPLIED - ML GROUP 7

RARES MEDELET

BARIS EROGLU

BOGDAN SANDOIU



MODEL USAGE

The VehicleNet workflow begins when a user records a short clip of engine noise and then selects the segment containing only the pure engine sound and converts it into a .wav file. That audio is fed into our preconfigured Docker container, ensuring a consistent environment for every analysis. The .wav file features are sent to our inference API, which preprocesses the sound and returns a json where you can see the classified class.

PROJECT NOVELTY

There are a lot of general classifiers that handle hundreds of labels, but ours is tailored specifically to engine sounds. The classic approach is to use an RNN or CNN architecture, but we wanted to combine them, to observe if we get a better performance in terms of classification.

VehicleNET

PREPROCESSING

- Downloading the raw dataset of engine sounds.
- Splitting the data into train/valid/test.
- Finding the highest sampling rate of each class.
- Resample all of the classes at the same sampling rate.
- Noise reduction by trimming silence parts of the audio file.
- Spectograms extraction from audio samples, which serves as input for CNN.
- Manually extracted features from audio samples, which serve as input for RNN.
- Apply dimensionality reduction (PCA) to manually extracted features.

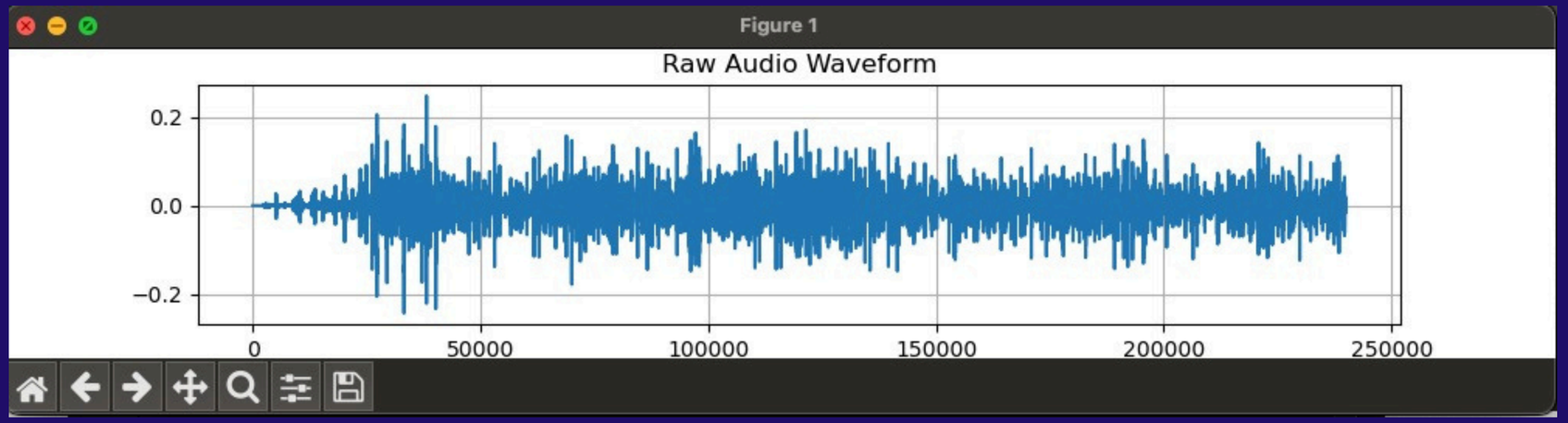


Figure 1
Raw Audio Waveform
0.2
0.0
−0.2
0
50000
100000
150000
200000
250000

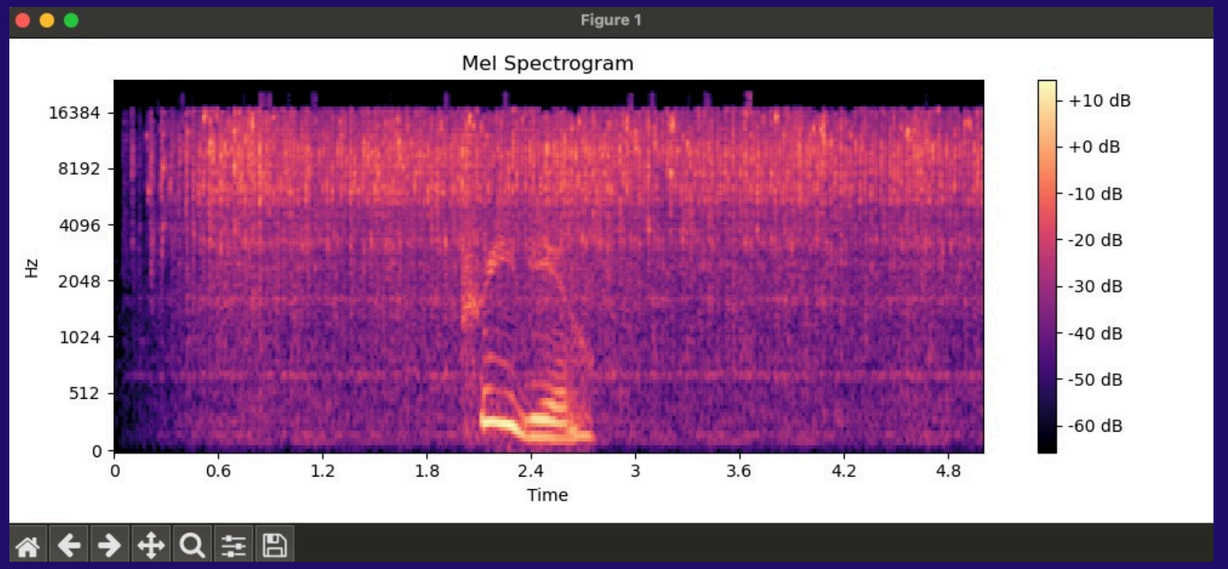


Figure 1
Mel Spectrogram
16384
8192
4096
2048
1024
512
0
Hz
0
0.6
1.2
1.8
2.4
3
3.6
4.2
4.8
Time
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB

VehicleNET

MODELS

Baseline Model:

- RNN architecture:
  - Good approach to learn long-term dependencies of time series input (manually extracted features).
- CNN architecture:
  - Good approach to learn spectrograms.

Main Model:

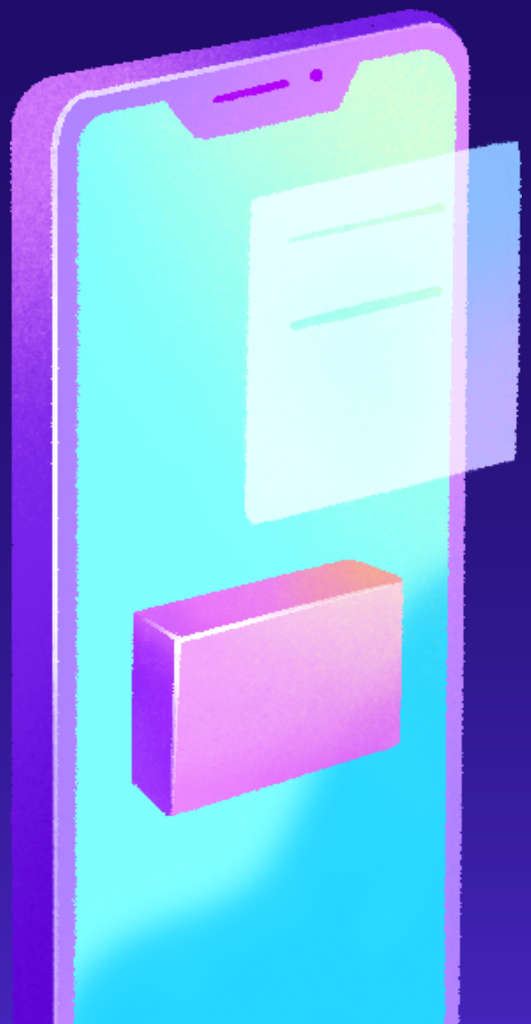
- RNN + CNN architecture:

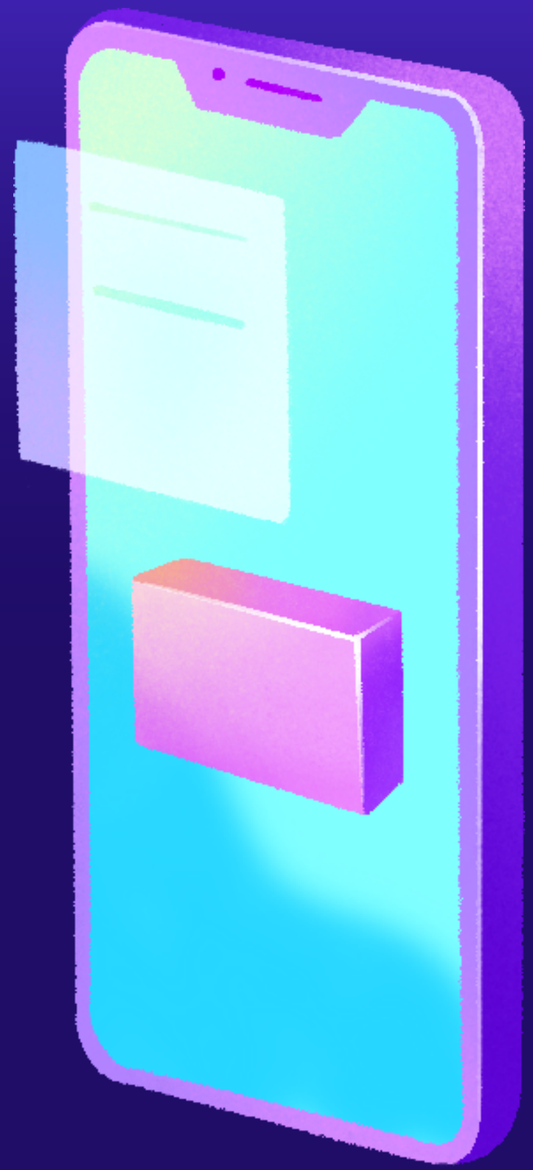
Regularization:

- We used dropout in order to regularize our model and avoid overfitting.

RESULTS

- We had better results than expected for the baseline model.
- There is no significant difference between the baseline and the main model.
- Metrics used:
  - Confusion Matrix
  - F1 Score
- Train/Valid loss plots.



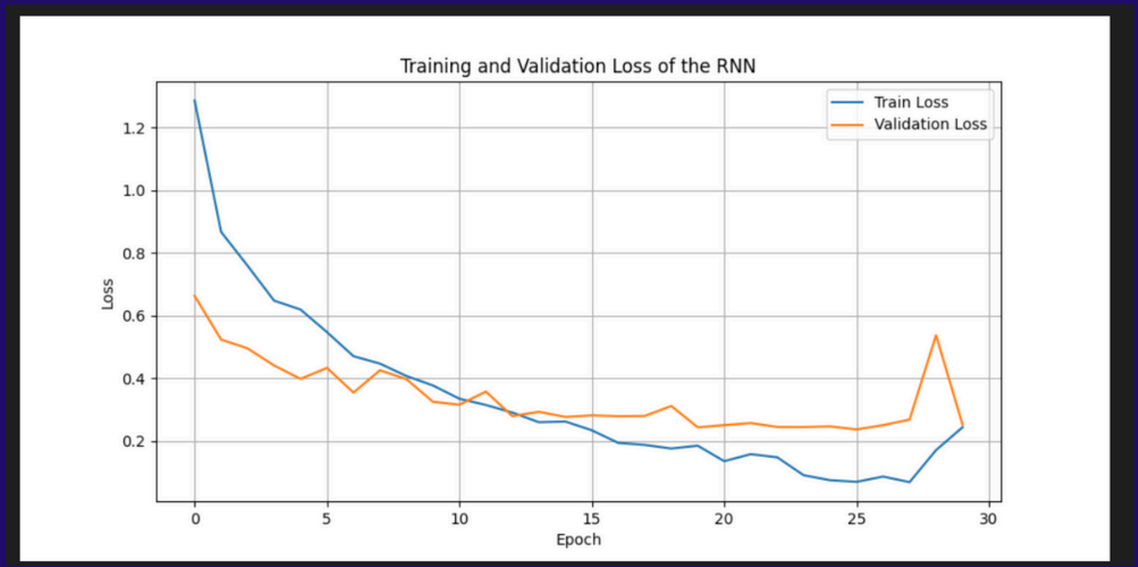


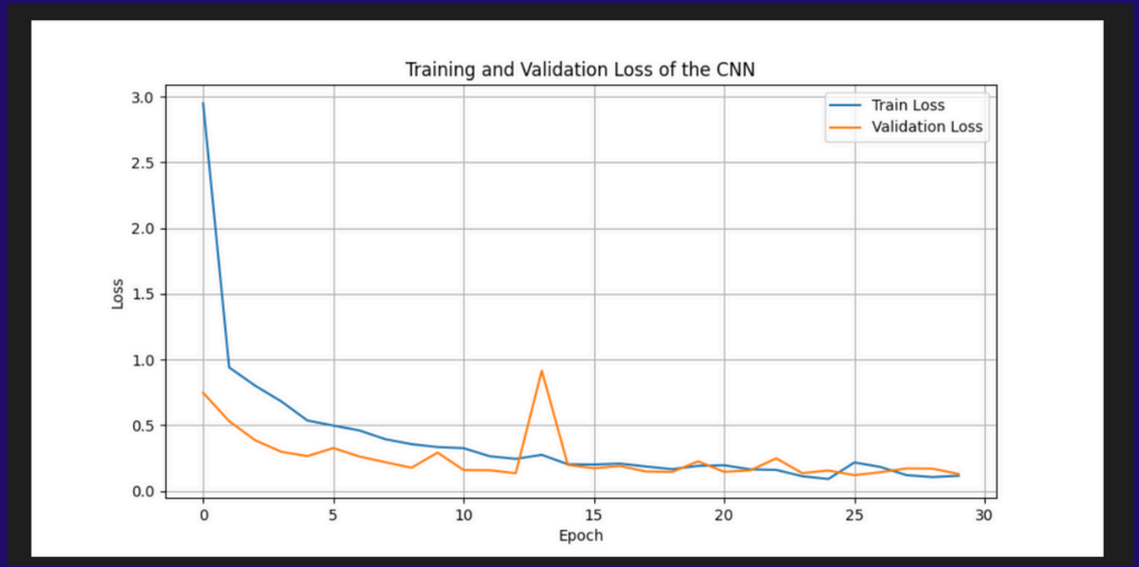
CNN: accuracy			0.97	1426
macro avg	0.92	0.94	0.93	1426
weighted avg	0.97	0.97	0.97	1426

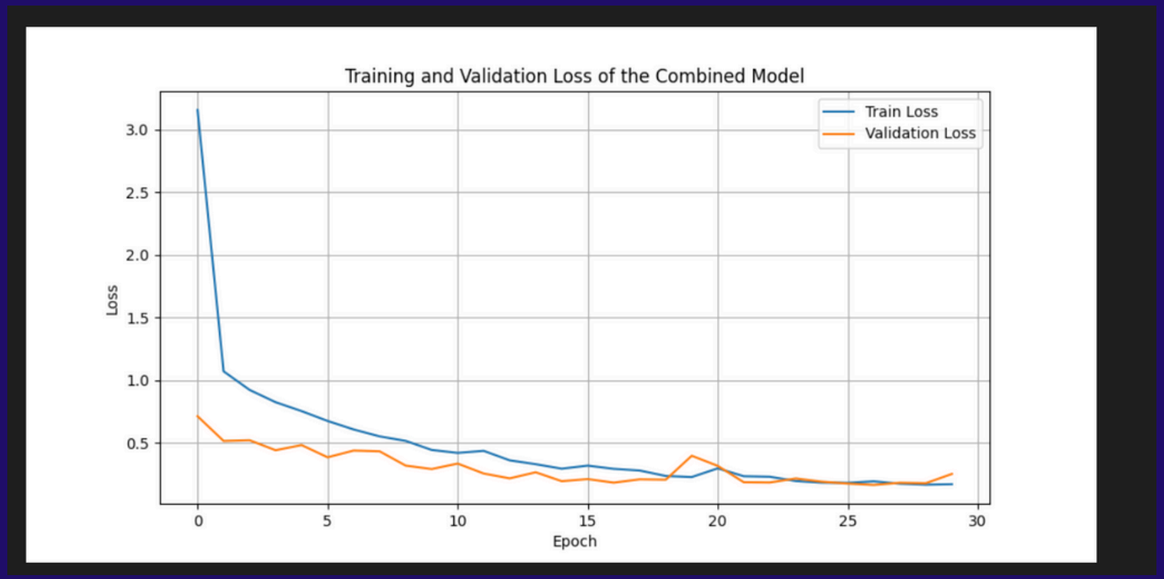
accuracy			0.94	1426
macro avg	0.86	0.89	0.87	1426
weighted avg	0.94	0.94	0.94	1426

Combined Model: accuracy			0.92	1426
macro avg	0.82	0.89	0.85	1426
weighted avg	0.94	0.92	0.92	1426

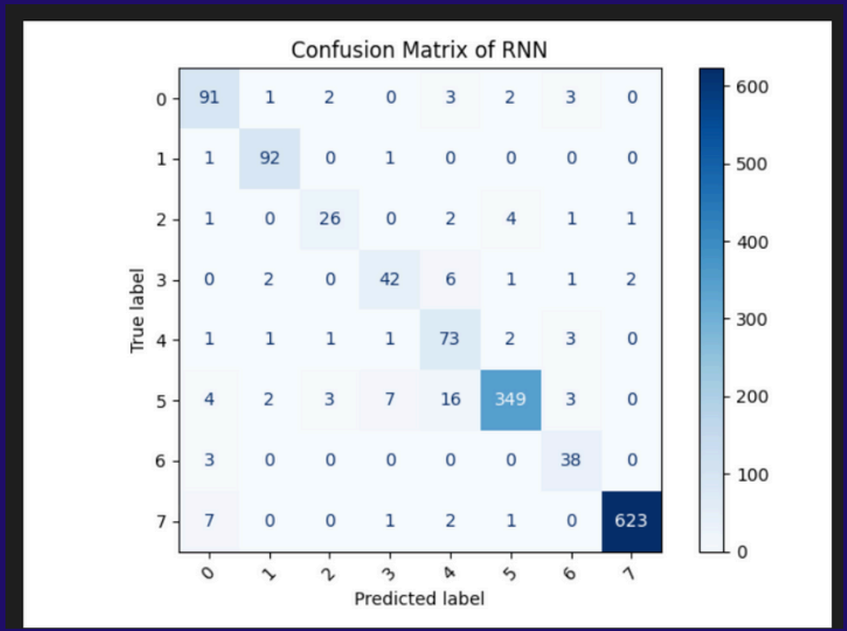
OBSERVING OVERFITTING

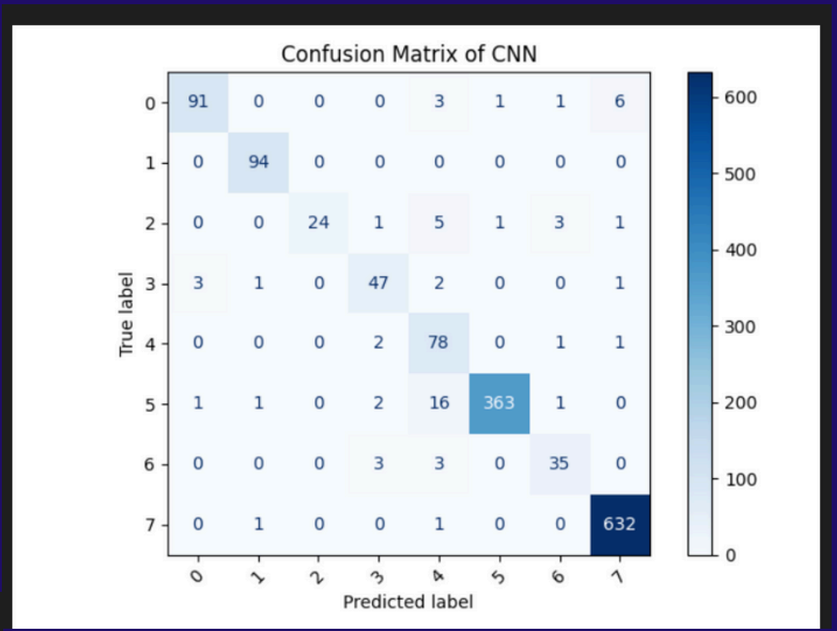


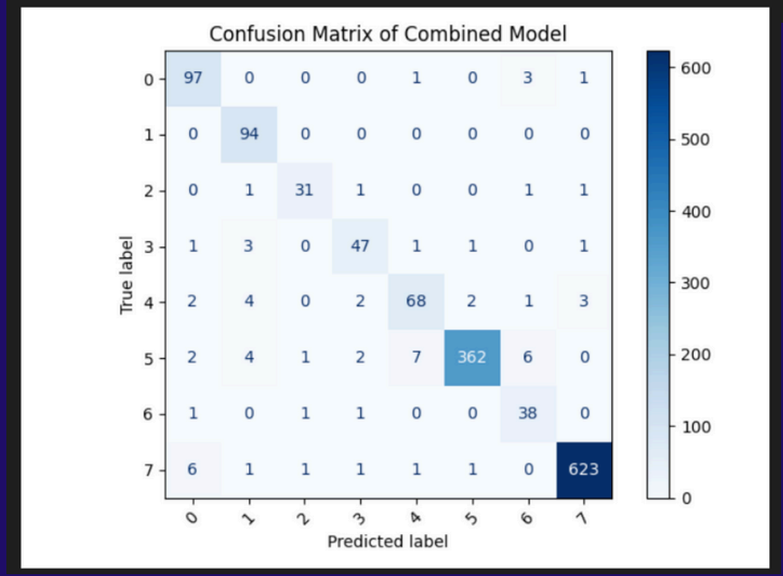




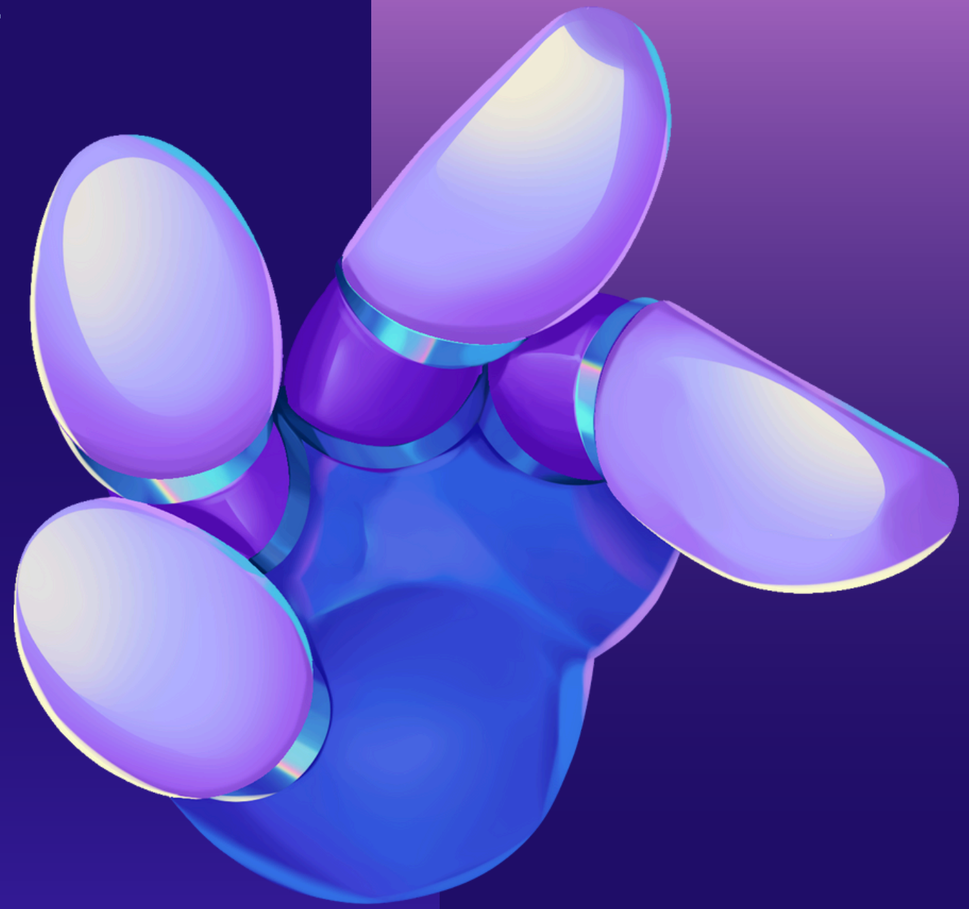
CONFUSION MATRICES







VehicleNET



...

Professionalism

Phase 01

DOCKER !

Phase 02

BRANCHES !

Phase 03

CLEAN CODE !

EXTRAS

- Address **data imbalance.**
- Apply dimensionality reduction.
- Explore meaningfully **different types of models.**
- Feature engineering.
- A nice streamlit demo.

MORE EXTRAS

- Containerization with **Docker.**
- Evaluation with real data.

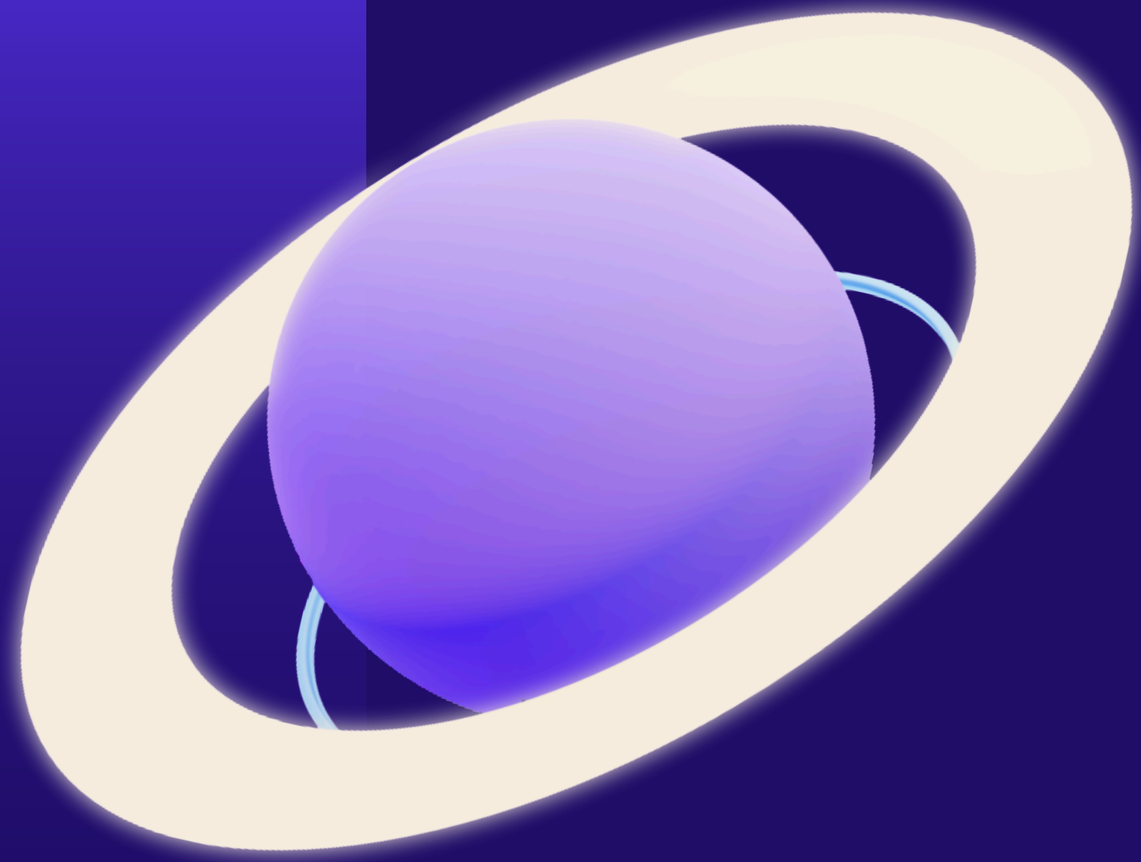
Main problems encountered:

1. Sound similarity between classes
2. Noise, noise, noise, a lot of noise

PROBLEMS

THE FUTURE IS PROMISING

QUESTIONS?



THANK YOU !