IBM Advanced Data Science Capstone Project

Garbage Classifier

Video Presentation

Filippo Bertelè

Waste management

Status quo

- About 90% of global waste end up in landfills or in the oceans
- Not sustainable
- Pollution is growing exponentially
- Excessive use of raw materials

Recycling is a possible solution

Recycling

Current problems

- Trash usually comes all mixed
- Recycling works best if each material is separated from the others
- The separation process can be slow and require a lot of manpower
- Current machines can mostly separate only metals

Software solution

The idea

- Fast identification of different materials
- With fine-tuning it can reach really high accuracy
- Easy implementation with already existing hardware
- Almost no maintenance required
- Easily scalable

Software solution

What is it ?

- Image classification software
- Approx 80% accuracy
- Recognition of 6 different materials

1. *Plastic*
2. *Cardboard*
3. *Paper*
4. *Metal*
5. *Glass*
6. *Trash*

The Dataset
What it sees

- Contains 2537 images
- RGB, 3 channels
- Original shape: 512 x 384
- Jpeg format
- 43.4 MB in size
- 6 different classes
- Hand picked and labeled

https://www.kaggle.com/asdasdasasdas/garbage-classification


Glass

Cardboard

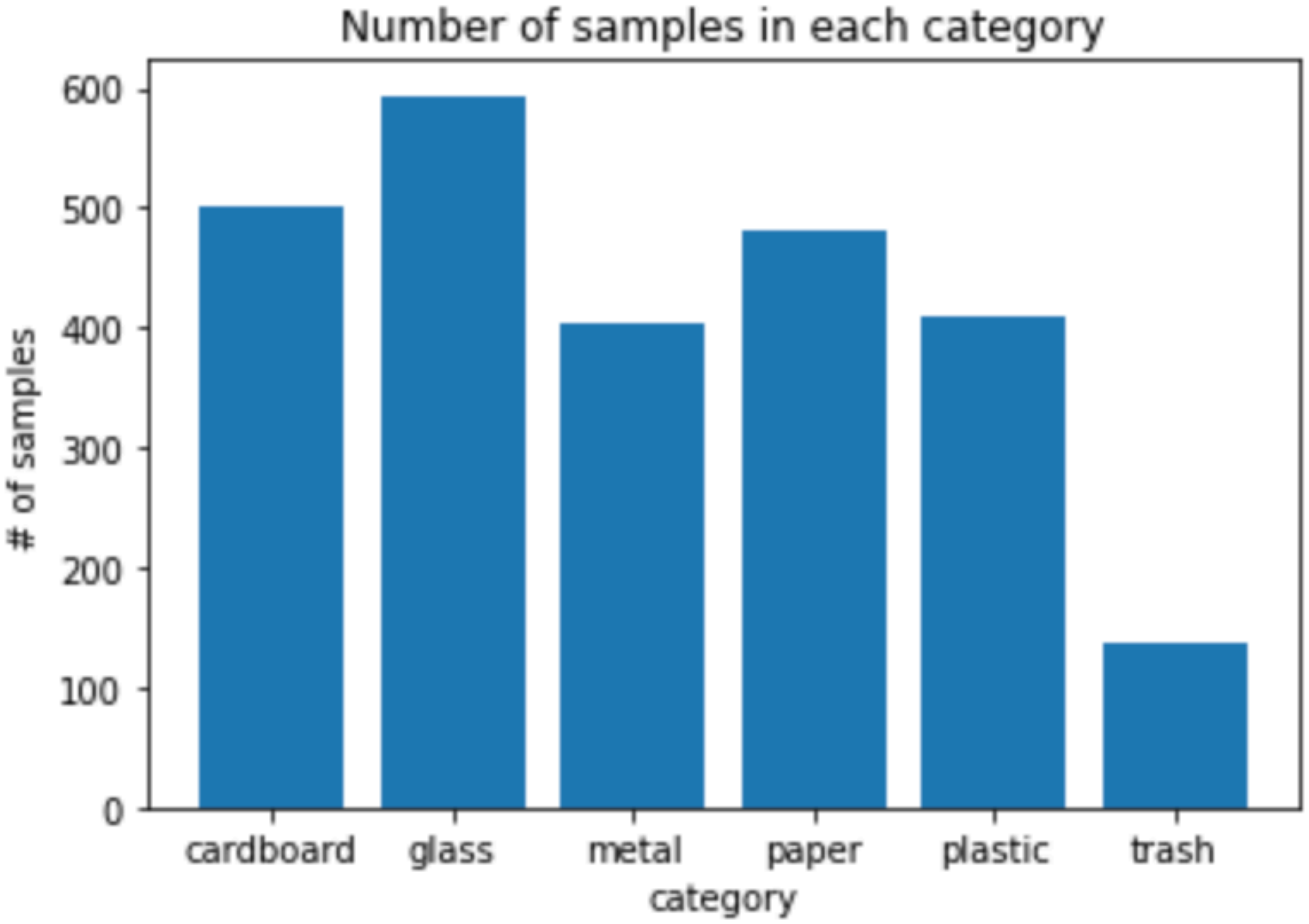

Plastic


Metal

Paper


Trash

The Dataset
Data exploration



- The sample distribution in the dataset is not balanced

Feature engineering

Normalization

- Images consists of a tensor with values ranging from 0 to 255 and are restricted to be bounded between 0 and 1
- Normalization can increase the performance of the model by preventing vanishing gradient and exploding gradient problems
- Two different models are trained:
    - The first one with non normalized data
    - The second one with normalised data
    - Finally the models' performance are compared

Feature engineering

Data augmentation

- The dataset is relatively small and 10% of it is resaved for the validation phase
- To increase the generality of the model the training samples are augmented by means of:
    - Rotation
    - Shear
    - Flips
    - Shifts
    - Zooms

Algorithms

Logistic Regression

- The first model is a simple Linear Regression model
- SoftMax activation function

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_6 (Flatten)          (None, None)              0
_________________________________________________________________
dense_10 (Dense)             (None, 6)                 2654214
=================================================================
Total params: 2,654,214
Trainable params: 2,654,214
Non-trainable params: 0
_________________________________________________________________
```

Algorithms

Convolutional Neural Network

The second model is a Convolutional Neural Network:

- 4 Convolutional layers
- 3 Fully Connected layers
- ReLu and Softmax activation functions
- MaxPooling and Dropout layers for dimensionality reduction and to reduce overfitting

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 384, 384, 16)	448
max_pooling2d (MaxPooling2D)	(None, 192, 192, 16)	0
conv2d_1 (Conv2D)	(None, 192, 192, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 96, 96, 32)	0
conv2d_2 (Conv2D)	(None, 96, 96, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 48, 48, 64)	0
conv2d_3 (Conv2D)	(None, 48, 48, 32)	18464
max_pooling2d_3 (MaxPooling2	(None, 24, 24, 32)	0
flatten_1 (Flatten)	(None, 18432)	0
dense_1 (Dense)	(None, 64)	1179712
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 6)	198

Total params: 1,224,038
Trainable params: 1,224,038
Non-trainable params: 0

Training

LR model

- Number of epochs: 20

CNN model

- Number of epochs: 80

- The monitored metrics are : training accuracy, training loss, validation accuracy and validation loss
- Loss function is: Categorical CrossEntropy
- The model with the highest validation accuracy is saved at each epoch
- To prevent overfitting the learning rate is adjusted to 0.2 x lr after 8 epochs without validation accuracy improvement

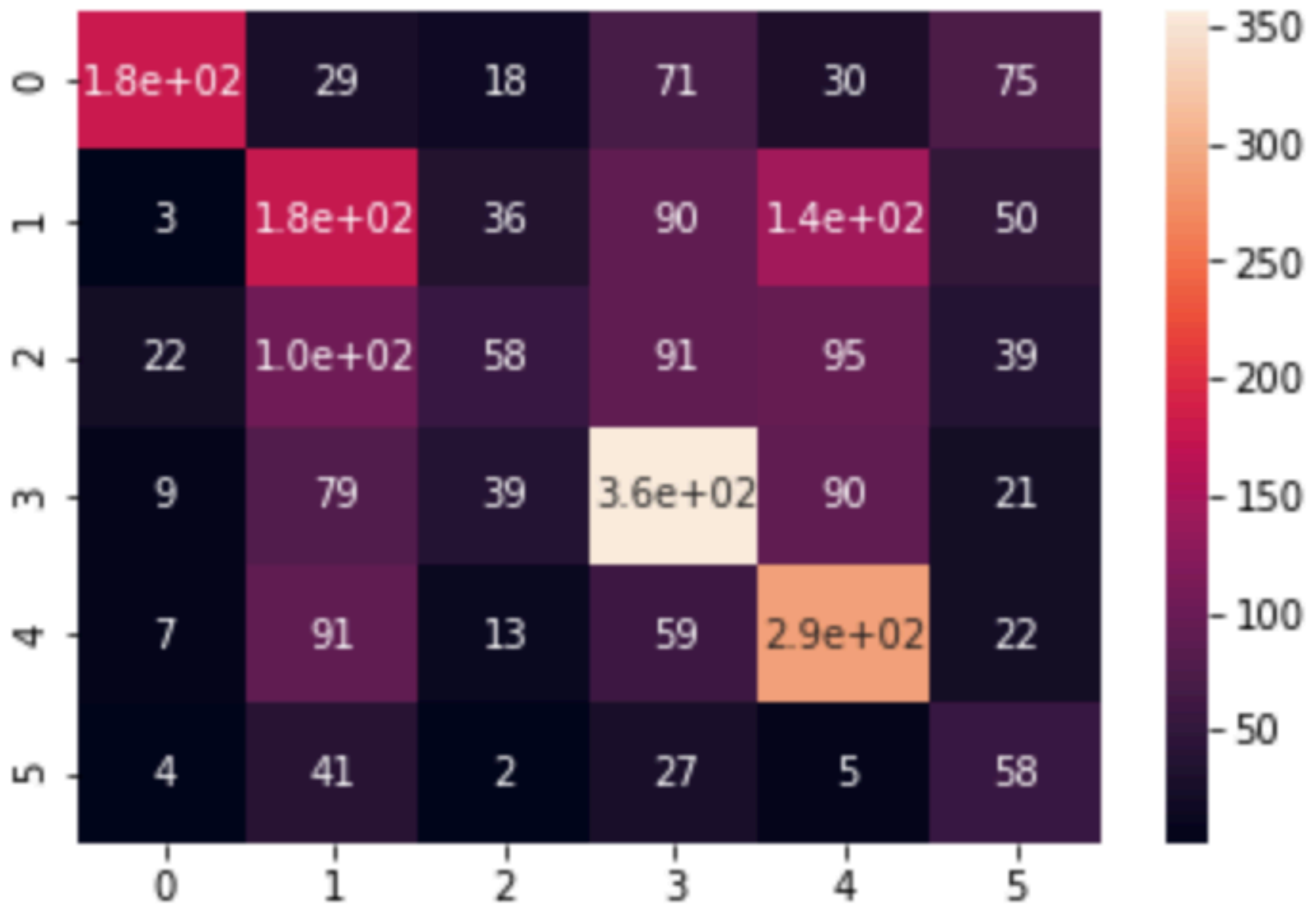
Evaluation
Performance indicators

- **Accuracy**:
  - Accuracy represents the percentage of correct predictions made on the test set
- **Confusion Matrix**
  - Visual representation of the performance on each class
- **F1 Score**
  - Weighted average between accuracy and sensitivity

Evaluation

LR Results

- The Linear Regression model performed well, considered its size and complexity
- Accuracy : > 35%
- F1 Score : 0.22



Evaluation

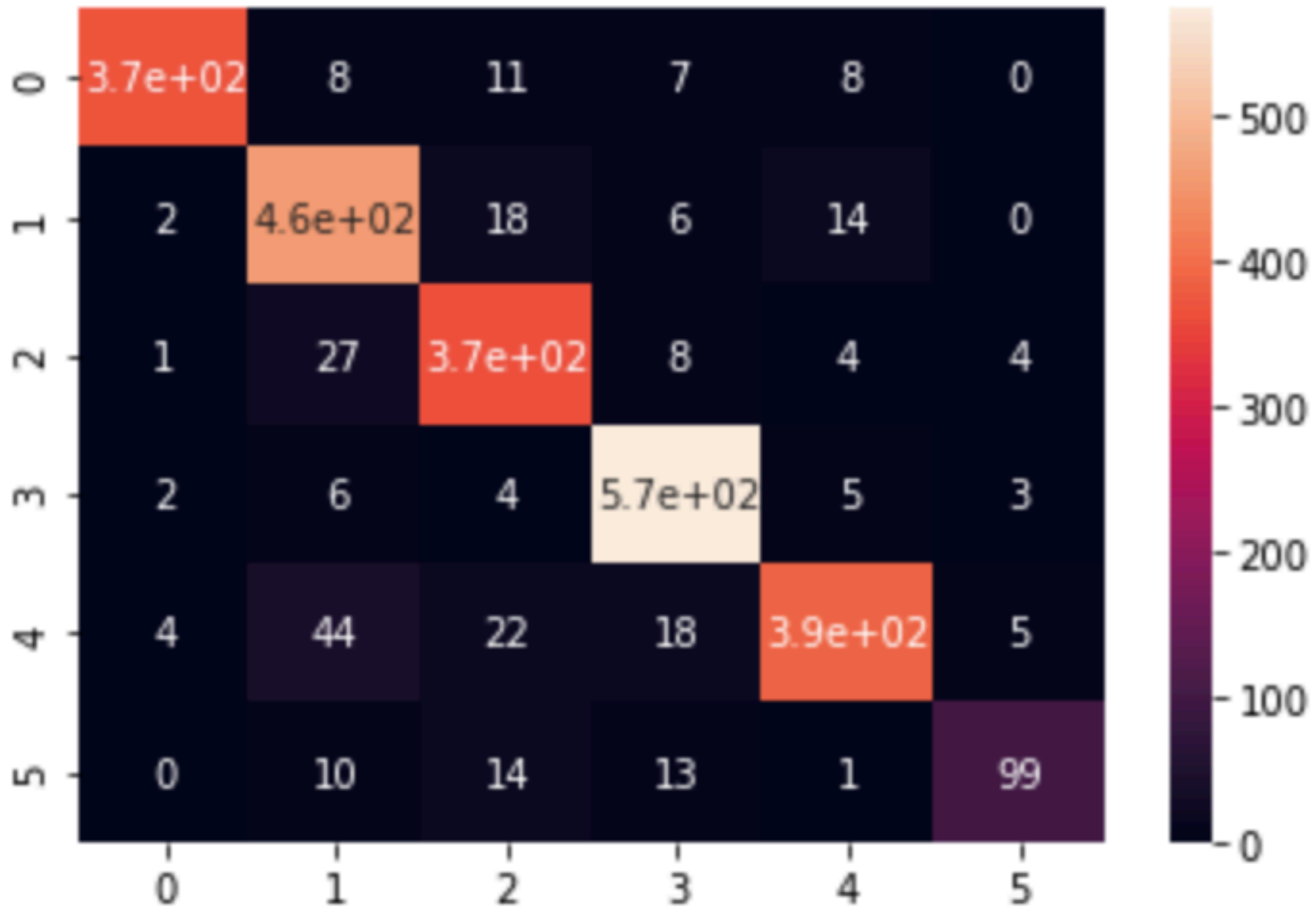
CNN Results

- As expected the CNN without normalization performs much better than the LR model
- Accuracy : 71%
- F1 Score: 0.83

Evaluation

Normalized CNN Results

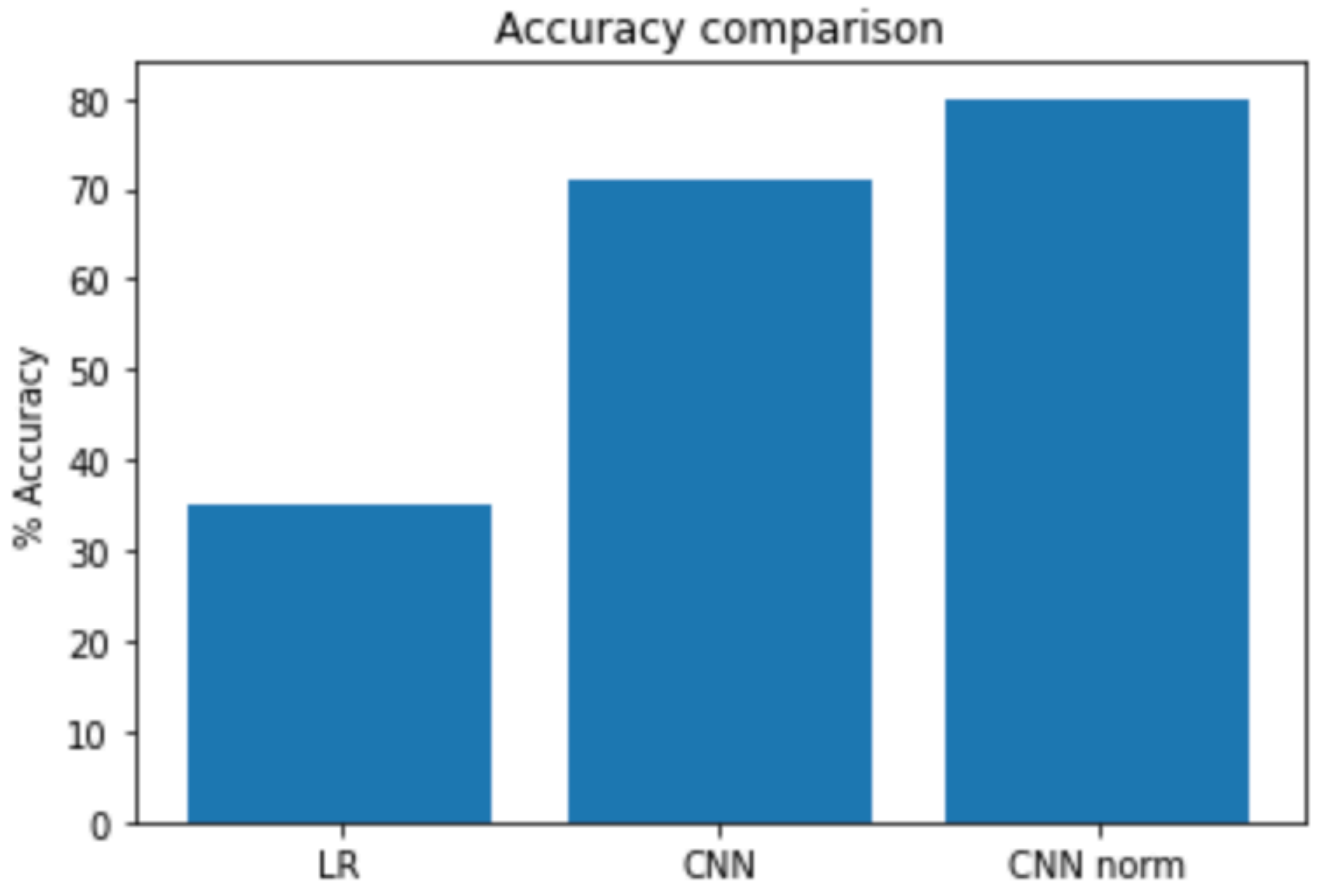
- The CNN with normalization perform best with highest score in each monitored metric
- Accuracy : 80%
- F1 Score: 0.89



Results

Accuracy comparison

- The accuracy of the CNN with normalized data is 80%
- It shows a 12% increase with respect to the CNN without normalization
- And a 128% increase with respect to the Logistic Regression model



Results

F1 Score comparison

- The F1 Score of the CNN with normalized data is 0.89
- It shows a 7% increase with respect to the CNN without normalization
- And a 404% increase with respect to the Logistic Regression model