Veritas AI: (CIFAR-10)

[Vivhaan Oswal, Varnika Kothari, Siddhanth Ganesh, Annamalai Muthiah]
[23/5/25]

Motivation

S - Situation
T - Task
A - Action
R - Result

Research question

How accurately can a convolutional neural network trained on the CIFAR-10 dataset classify images for use in self-driving car perception systems?

Project setting

The data is from Canadian Institute For Advanced Research (CIFAR-10) dataset. It has 60 000 photographs from 10 groups, which are airplane, ship, dog, cat, frog, horse, deer, truck, automobile and bird. We are to use CNN to try and properly classify each image to its respective class

Example of the raw images

S - Situation
T - Task
A - Action
R - Result

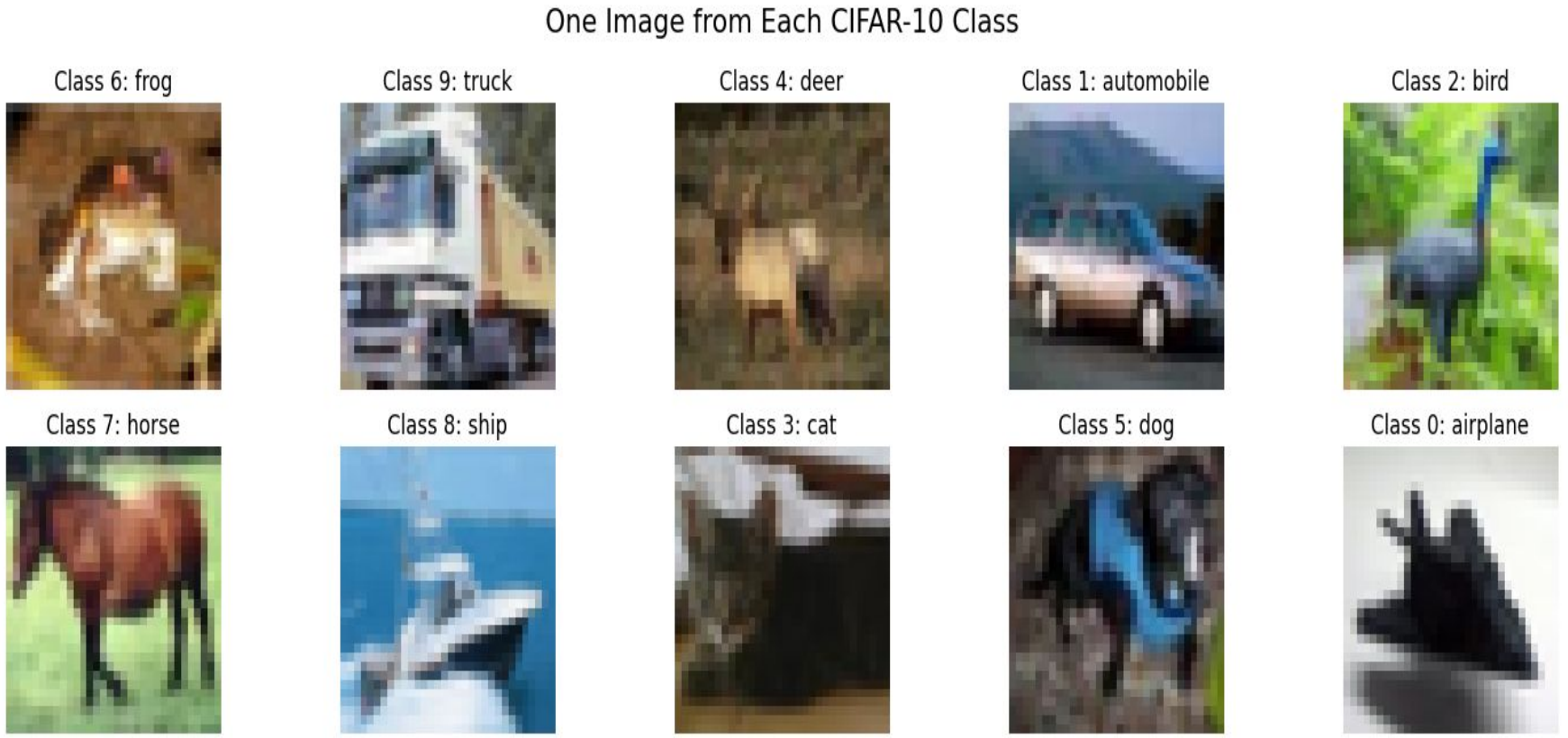
The CIFAR-10 dataset

- The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.
- There are 50000 training images and 10000 test images.
- Researchers at the University of Toronto compiled CIFAR-10 in 2009.

Exploratory Data Analysis(EDA)

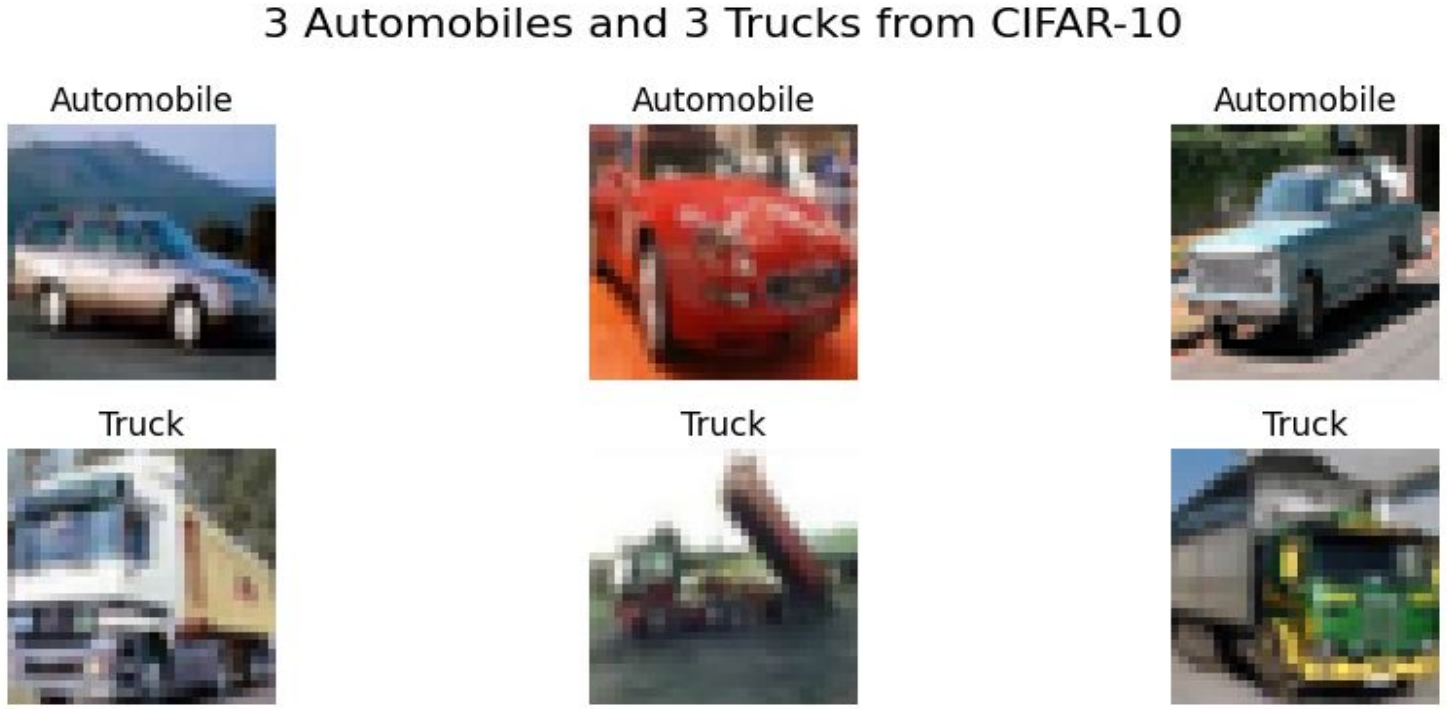
S - Situation
T - Task
A - Action
R - Result

For the CIFAR-10 data, it can be classified into 10 different groups as shown:



Similarities and Differences(EDA)

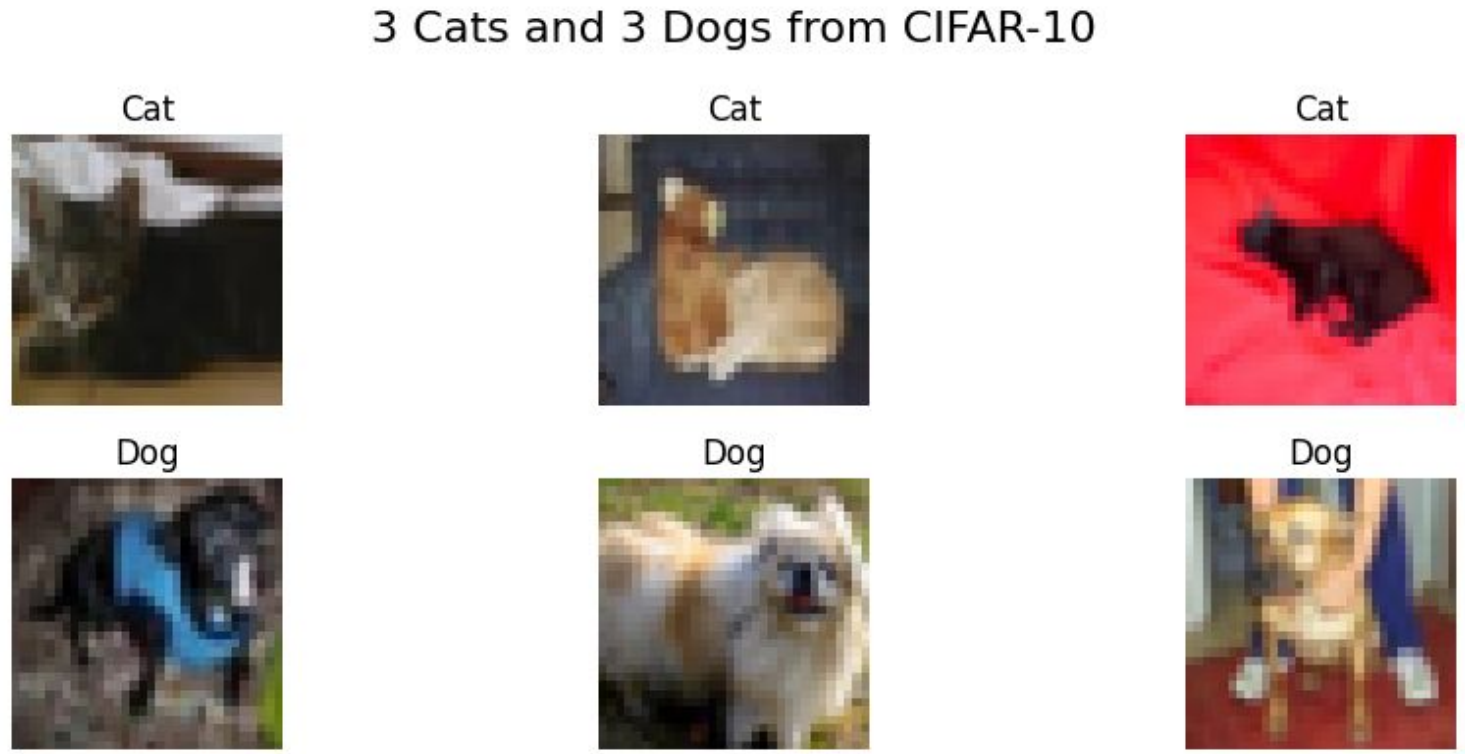
S - Situation
T - Task
A - Action
R - Result



Name	Distinct characteristics	Similar with + reason
Automobile and Truck	• A truck has a more boxy build while an automobile has a more streamlined	- Kind of similar to a truck due to sharing wheels and a metal body - Some cars do share a boxy look as well

Similarities and Differences(EDA)

S - Situation
T - Task
A - Action
R - Result



Name	Distinct characteristics	Similar with + reason
Cat and dog	• A dog has a snout like face while a cat has a rounded face	- similar with the dog due to the relatively similar shape/size and ears

Similarities and Differences(EDA)

S - Situation
T - Task
A - Action
R - Result

Name	Distinct characteristics	Similar with + reason
Deer and Horse	• A deer has antlers while the horse has a mane	- due to the poor rendering both have a similar body structure and in some cases it's hard to identify the antler

Exploratory Data Analysis

S - Situation
T - Task
A - Action
R - Result

Summarized observations

Name	Distinct characteristics	Similar with + reason
Automobile and Truck	• A truck has a more boxy build while an automobile has a more streamlined	- Kind of similar to a truck due to sharing wheels and a metal body
Cat and Dog	• A dog has a snout like face while a cat has a rounded face	- similar with the dog due to the relatively similar shape/size and ears
Deer and Horse	• A deer has antlers while the horse has a mane	- due to the poor rendering both have a similar body structure and in some cases it's hard to identify the antler

Baseline Model and Results

S - Situation
T - Task
A - Action
R - Result

The code

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(60, (3, 3), padding="same", activation="relu",
input_shape=(32, 32, 3)),
    tf.keras.layers.Conv2D(40, (3, 3), padding="same", activation="relu"),
    tf.keras.layers.Flatten(),
    #tf.keras.layers.Dense(160, activation="relu"),
    tf.keras.layers.Dropout(0.7),
    tf.keras.layers.Dense(160, activation="relu"),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(80, activation="relu"),
    tf.keras.layers.Dense(40, activation="relu"),
    tf.keras.layers.Dense(10, activation="softmax"),
])
```

- For the baseline model we used a convolutional neural network consisting of six layers.
- The CNN's architecture was built upon using a combination of CNN layers, normal neural network layers, activation functions (including relu and softmax) and dropout functions.
- The model was run for 12 epochs and we got an accuracy of around 67.30% on the test data and 75.18% on training data. This portrays that the model was slightly overfit.

Baseline Model and Results

S - Situation
T - Task
A - Action
R - Result

Misclassified Photos examples:

shares a large structure

random error?

looks like a floating ship

Random error?

Random error?

mistaken for a van due to boxy shape?

Random error?

resembles a flying bird

Advanced Model and Results

S - Situation
T - Task
A - Action
R - Result

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same', input_shape=(32, 32, 3)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3), padding='same' ))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3), padding='same' ))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.35))

model.add(Conv2D(128, (3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Conv2D(128, (3, 3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.4))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
```

- The baseline model was simple and didn't include helpful techniques, which make training more stable. To improve it, we made the model deeper by adding more convolutional layers and MaxPooling layers. We also added methods like Dropout and Batch Normalization to prevent overfitting, so the model could learn better and make more accurate predictions.

- The advanced model performed better. We were able to top it to a 80.4%. It trained more smoothly and achieved higher accuracy on the test data compared to the baseline model. It was able to generalize better and make more accurate predictions.

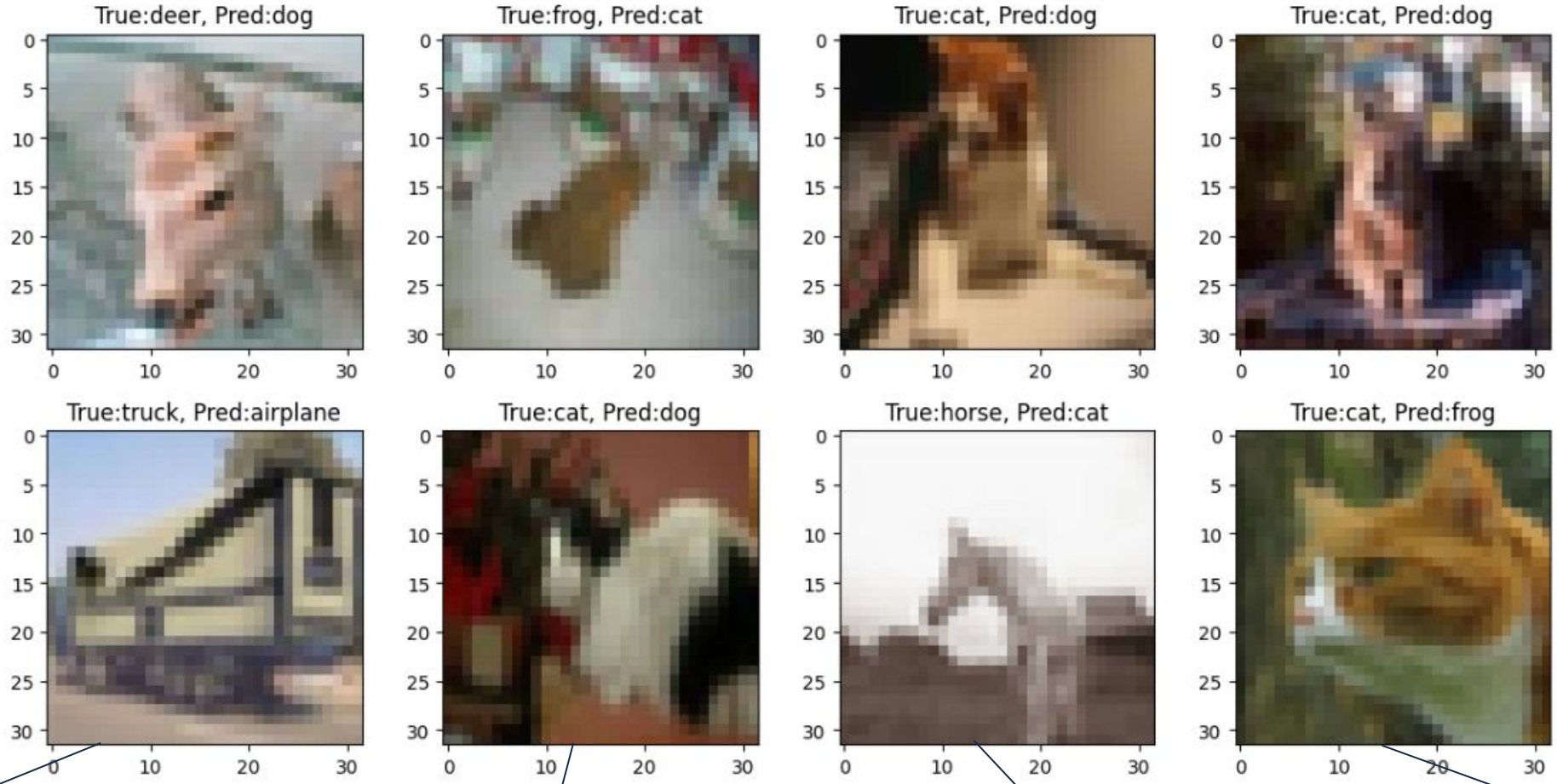
Misclassified photos for advanced model

Antlers not clearly visible

Low pixel quality

Cat's face is not visible

Low Pixel quality



Presence of sky like blue backdrop

Unclear image

Relative size is very small

Random error

Project Summary

S - Situation
T - Task
A - Action
R - Result

Our best model was a Convolutional Neural Network which consisted of 1 input layer, 15 hidden layers and 1 output layer. It was a combination of Dense layers, Batch Normalisation, Max Pooling and Dropout.

While programming and figuring out the best model architecture, we realised a couple of things-

1. Incorrect predictions were not only due to model errors but also because of low pixel quality or unclear images, as can be seen in the previous slides
2. Increasing number of epochs leads to an increase in accuracy upto a certain point post which it leads to a sudden fall
3. A high number of layers, or increased complexity of model architecture leads to a very high training accuracy but extremely low test accuracy (overfits the model)
4. The number of neurons in each layer should increase closer to the output layer to increase accuracy
5. An Increase in dropout value leads to faster execution

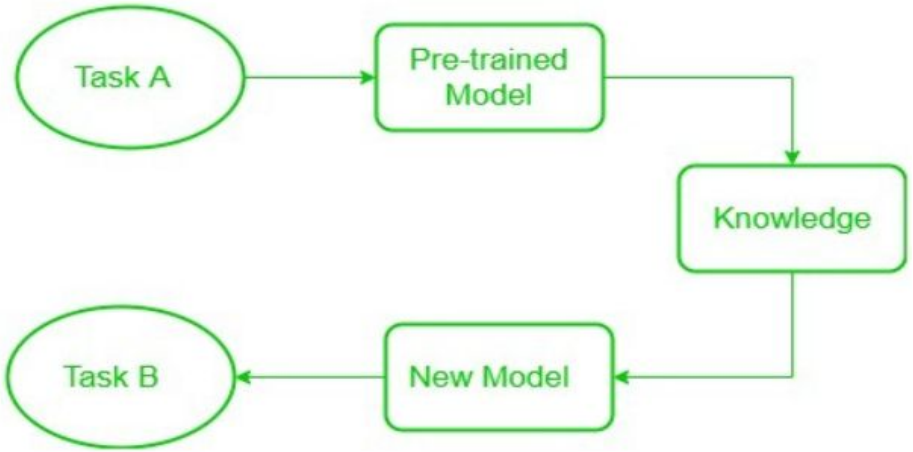
The highest accuracy achieved was 80.4% on the test data set with a relatively well tuned model which had 79.8% accuracy on the training set. This proves that the model was not under or over fit for the training data.

Future Scope For Enhancing The Model

S - Situation
T - Task
A - Action
R - Result

1. Transfer Learning

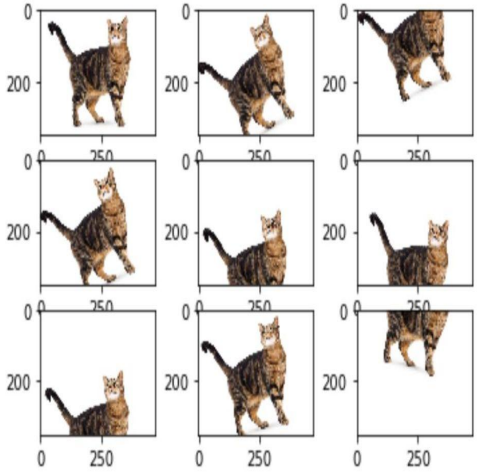
Transfer learning in machine learning is a technique where knowledge or features learned from a model on one task are reused on a related, but possibly different, task. This approach can significantly speed up training and improve performance. An additional benefit is that it reduces the quantity of data required to train the model.



2. Data Augmentation

Data Augmentation is a technique that artificially increases the size and diversity of a dataset by creating new data samples from existing ones.

We believe that using data augmentation will significantly expand the model's potential by increasing the number of training images. By creating modified versions of existing images, we can enhance the dataset without risking overfitting, allowing the model to learn more effectively



Future Use Cases Of Our Model

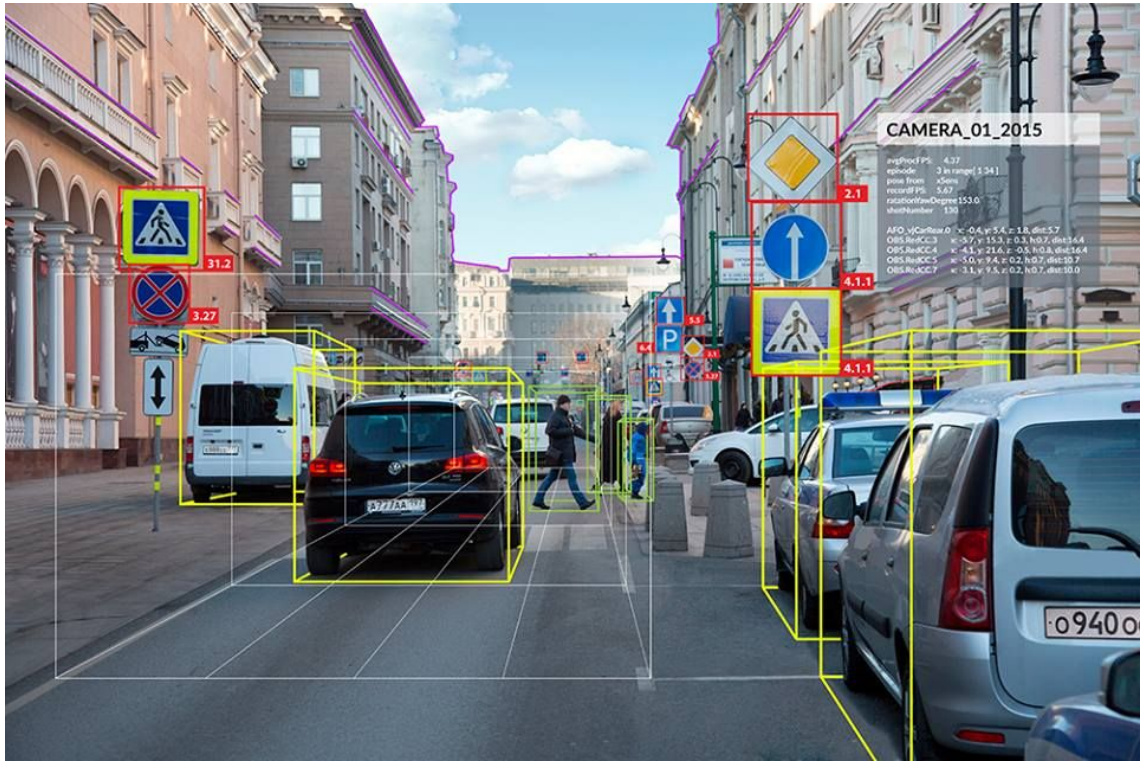
1. The main function of our model is to accurately scan the environment of self-driving cars in order to make sure they work efficiently.

2. Can be used by wildlife conservationists to study distribution of the different species in wildlife

3. Can be used in security and surveillance

4. Can be used in image editing software to identify key components in images

5. Can be used in manufacturing to do quality control



THANK YOU