Veritas AI: Image Recognition System Model

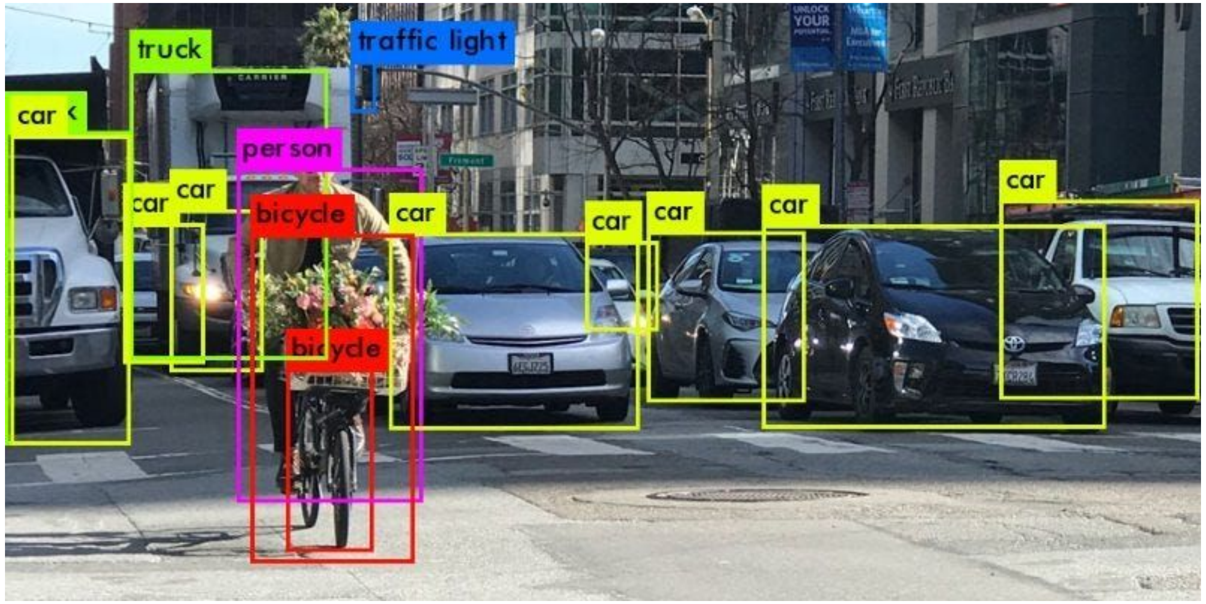
[Micah Lenus. Jackson Pulliam, Ayush Sadekar, Tom Lu, Vishal Veeramachaneni]

[07/28/2023]

Motivation for Your Project

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

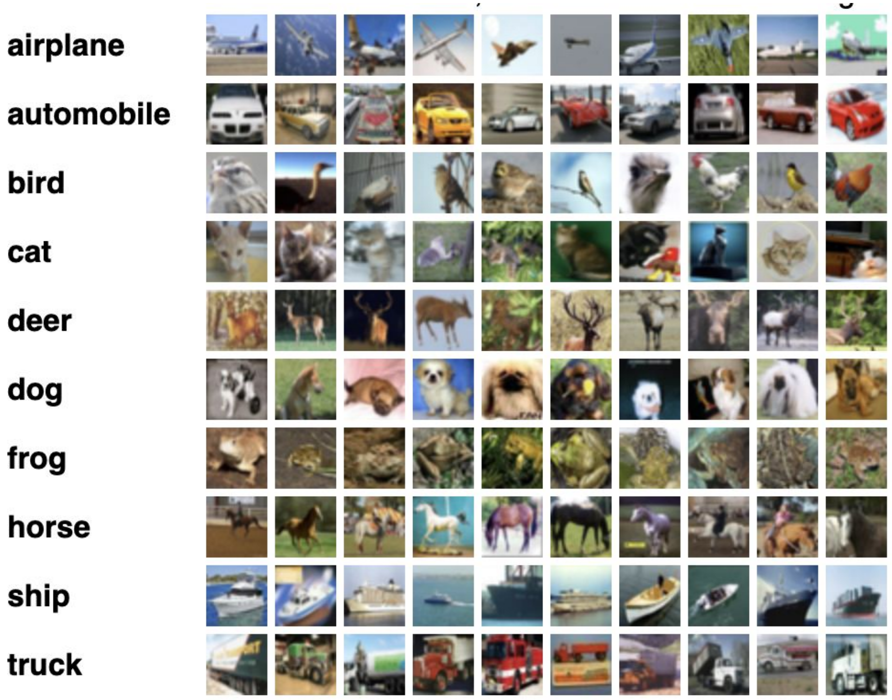
- **What is your research question?**
  Objective: Building an image recognition system which can identify different objects and patterns which may be more difficult for humans to identify.
- **Describe the setting for your project in 3 sentences**
  Building a self-driving system has to be done with an image recognition system. It is important for a vehicle to have a system which can identify images and recognize they are different so that the system can make decisions about how to drive accordingly. This includes recognizing many things which could be seen while driving, such as other vehicles, important signs, pedestrians, and animals.



Data

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

- Dataset name: CIFAR-10
- Benchmark dataset used in the field of computer vision and machine learning
- Dataset contains 60,000 pictures
- each picture is colored, resolution is 32x32 pixels
- Data collected by researchers at the Canadian Institute for Advanced Research(CIFAR)
- Dataset divided into ten distinct classes
- Each class of picture is mutually exclusive

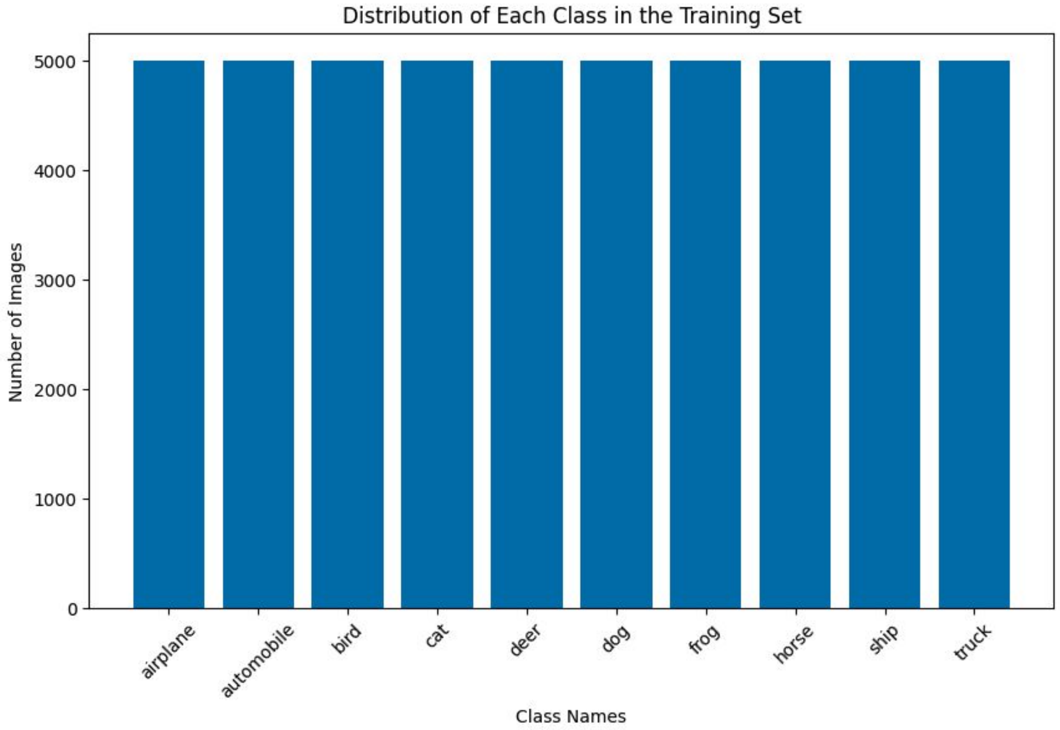


all ten classes of the dataset, as well as ten random pictures of each class

Exploratory Data Analysis

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

- 60,000 images in the data set
  - each image 32x32
  - 50,000 images used for training, 10,000 used for test data
- 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck
  - Each class was uniformly distributed in the training data — 5,000 images of each class




```
Total number of training examples: 50000
Total number of testing examples: 10000
Class distribution in the training set:
{'airplane': 5000, 'automobile': 5000, 'bird': 5000, 'cat': 5000, 'deer': 5000, 'dog': 5000, 'frog': 5000, 'horse': 5000, 'ship': 5000, 'truck': 5000}
```

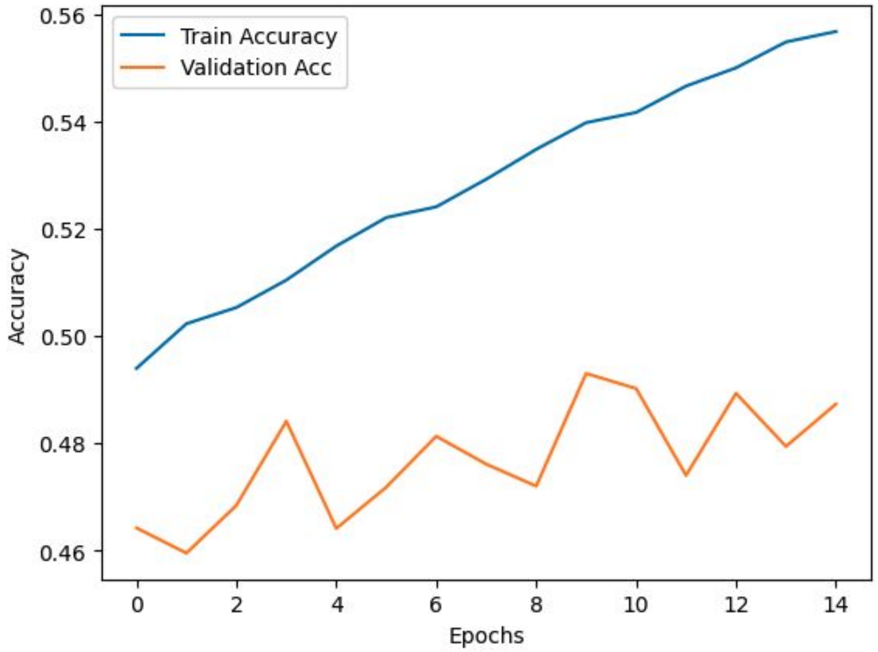
Exploratory Data Analysis(Pt. 2)

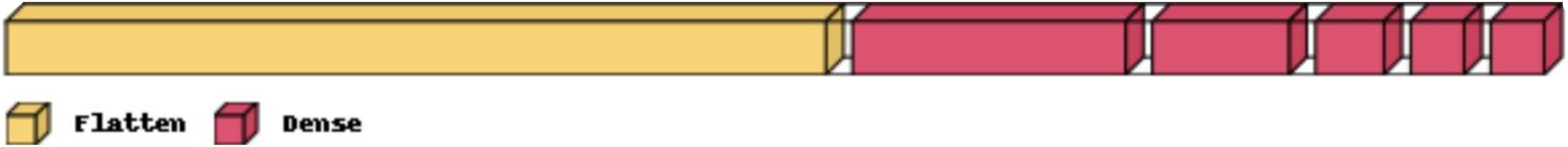
- Some of the images' features are more distinctive than others
- For example, it looks easier to identify automobiles and trucks. However, the images for cats and deers look a little more obscure. Images of horses with humans riding them may also confuse the model.

Baseline Model and Results

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

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_5 (Flatten)         (200, 3072)               0

 dense_24 (Dense)            (200, 1024)               3146752

 dense_25 (Dense)            (200, 512)                524800

 dense_26 (Dense)            (200, 256)                131328

 dense_27 (Dense)            (200, 128)                32896

 dense_28 (Dense)            (200, 10)                 1290

=================================================================
Total params: 3,837,066
Trainable params: 3,837,066
Non-trainable params: 0
_________________________________________________________________
```





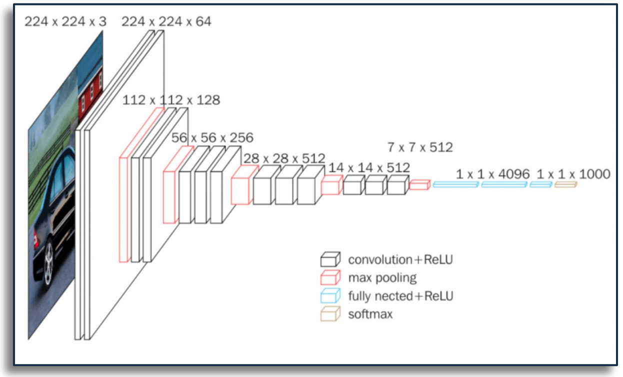
Improving the baseline model

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

$$\mathrm{BatchNorm}(x) = \gamma \frac{x - \mathrm{mean}(x)}{\sqrt{\mathrm{var}(x) + \epsilon}} + \beta$$

Convolution layers are layers within a neural network that add filters to data, allowing the network to see more directly parts of an image that are useful in classifying the image.

Batch Normalization is a method of improving deep neural network efficiency by normalizing the outputs of different layers within a network.

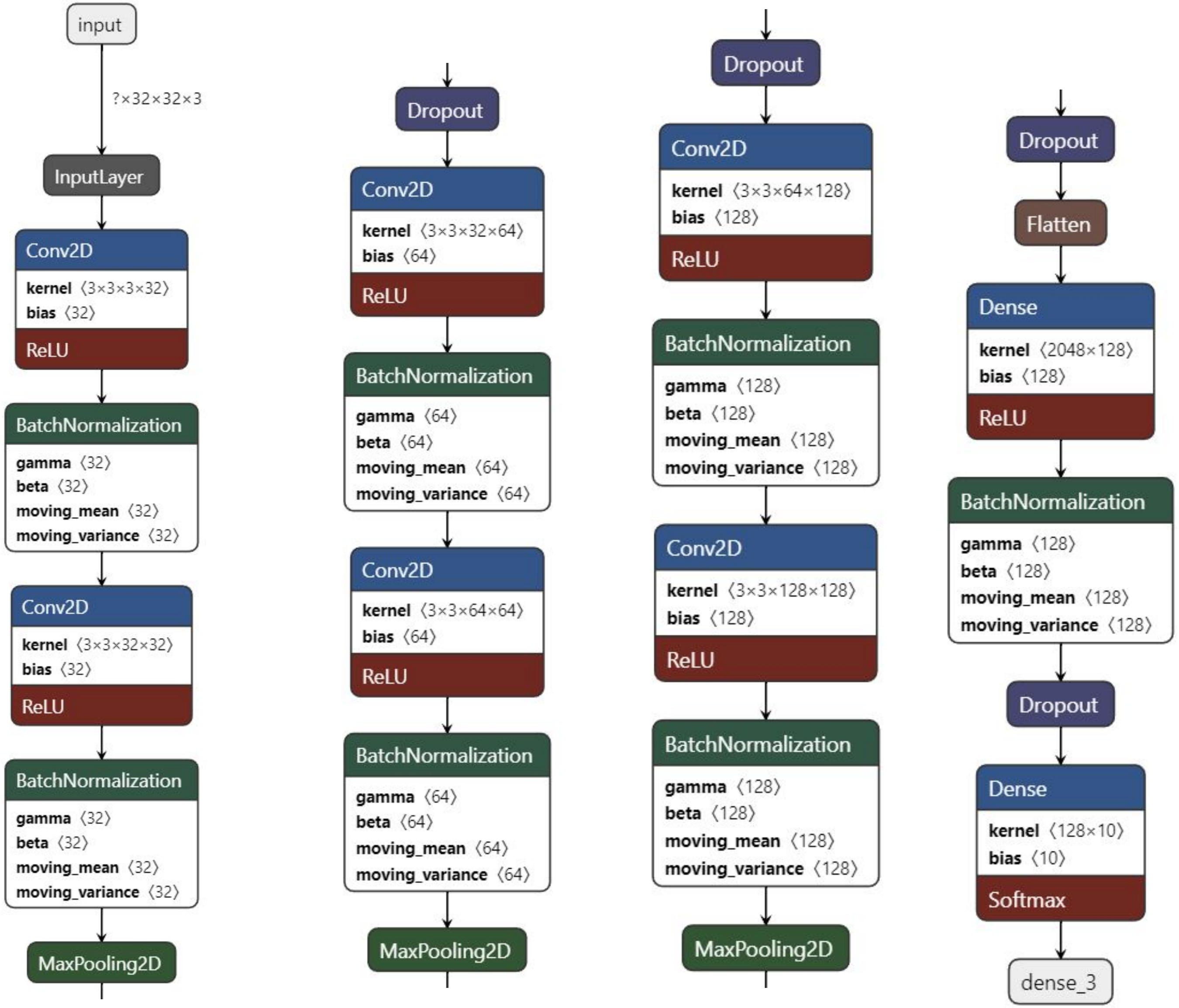


Dropout Layers randomly exclude certain input or hidden layer neurons from running during model training. This helps to prevent overfitting.

Epochs are iterations over the entire training data. The more epochs a model goes through, the more times the dataset is worked through.

Learning Rate determines the pace at which the model learns the dataset and updates its weights.

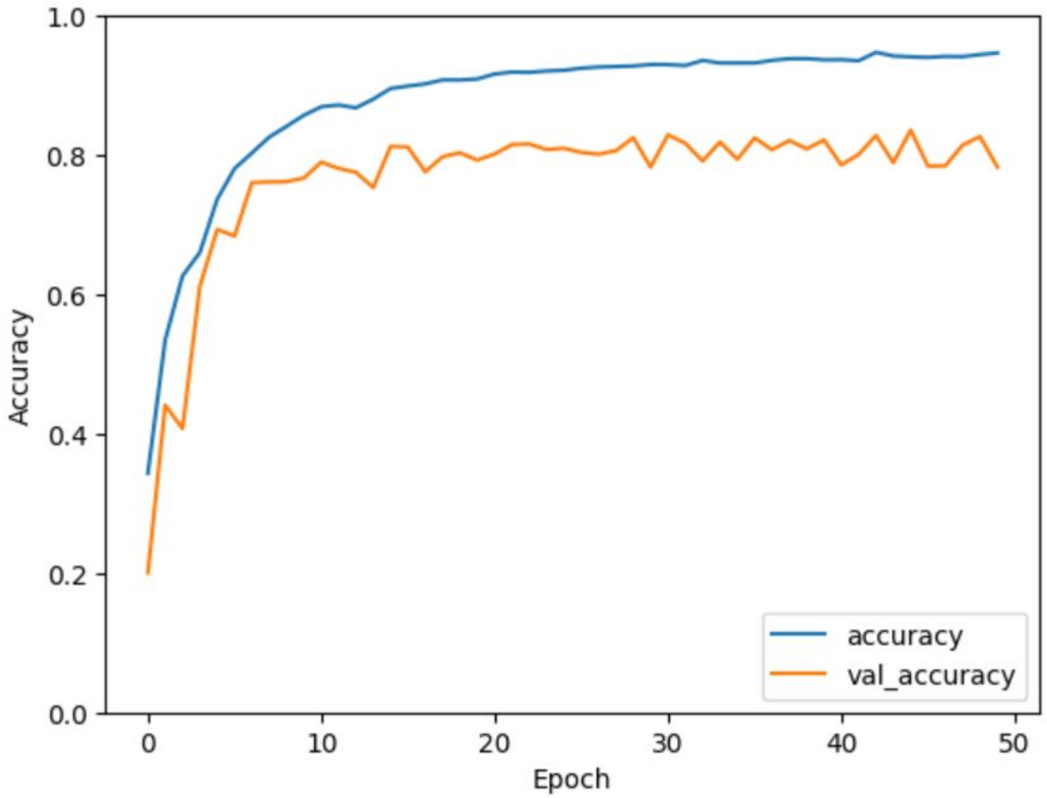
Improving the baseline architecture



Advanced Model and Results

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

Using some of these techniques, we were able to nearly double the performance of our model over the baseline model, getting a test accuracy of 82.08%.



Project Summary

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

- **Best Model**
  - The advanced model using a CNN was the best model
    - Training accuracy of 94.66%
    - Validation accuracy of 82.08%
- **What did we learn from the model?**
  - Some lessons learned from building the model include debugging and interpretation of loss and accuracy metrics.
  - The CNN model required more training time because of the deep layers

Future Work

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

- **Further Improving our Model**

  To further improve validation accuracy past its current maximum of 83%, data augmentation can be utilized. Also, hyperparameter tuning, such as varying the number of epochs or neurons per hidden layer, may improve the model. Finally, the learning rate can be reduced over time to help the model converge faster and at a more consistent rate.

Thank you!