# Deep Learning Final Project

## Objective

1. Classifying images within the CIFAR10 image dataset with a convolutional neural network.
2. Developing a strategy for an AI to play Blackjack.

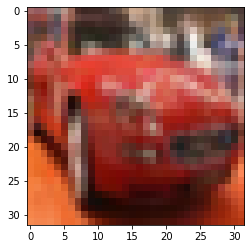
## Part 1 – CIFAR-10 Image Classification

### Dataset Information

CIFAR-10 is a subset of the larger 80 million image dataset that MIT collected in 2006. It used terms to automatically download images that matched with the term. This subset contains 60,000 total images. 50,000 of which are training image data and the remaining 10,000 are testing image data. The data can be further broken down into 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. There are 5,000 training images that correspond with each of those different classes.

The images provided are “tiny images”, so they are represented minimally with the number of pixels. It is difficult to manually determine what the images are. Some images are clearer than others, but overall the images are very pixelated.





|  |  |
| --- | --- |
| Number | Label |
| 0 | Airplane |
| 1 | Automobile |
| 2 | Bird |
| 3 | Cat |
| 4 | Deer |
| 5 | Dog |
| 6 | Frog |
| 7 | Horse |
| 8 | Ship |
| 9 | Truck |

### Data Cleaning

The first step in the data cleaning process was normalizing the image data. The data was converted to a float data type, then divided by 255. The 255 represents the number for RGB values.

After normalizing the image data, encodings needed to be applied to the labels. Because I am dealing with a multiclass classification problem, these labels needed to be converted into one hot vectors.

Because the image dataset already contained uniform 32x32 images, then there is not a need to reshape the images to a uniform resolution.

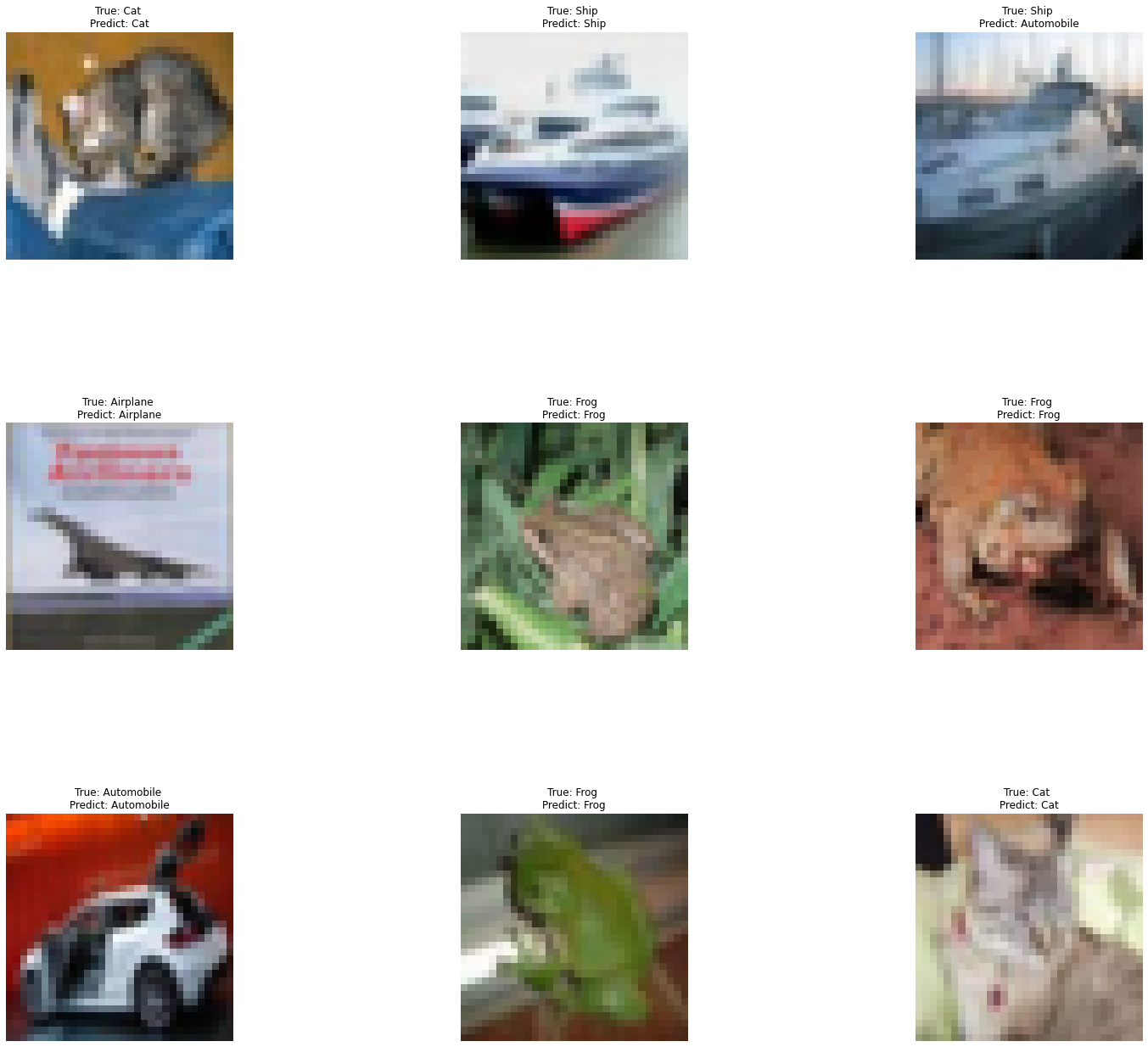
There wasn’t any data augmentation that was done on the data set as additional data could be pulled if the current image dataset wasn’t sufficient.

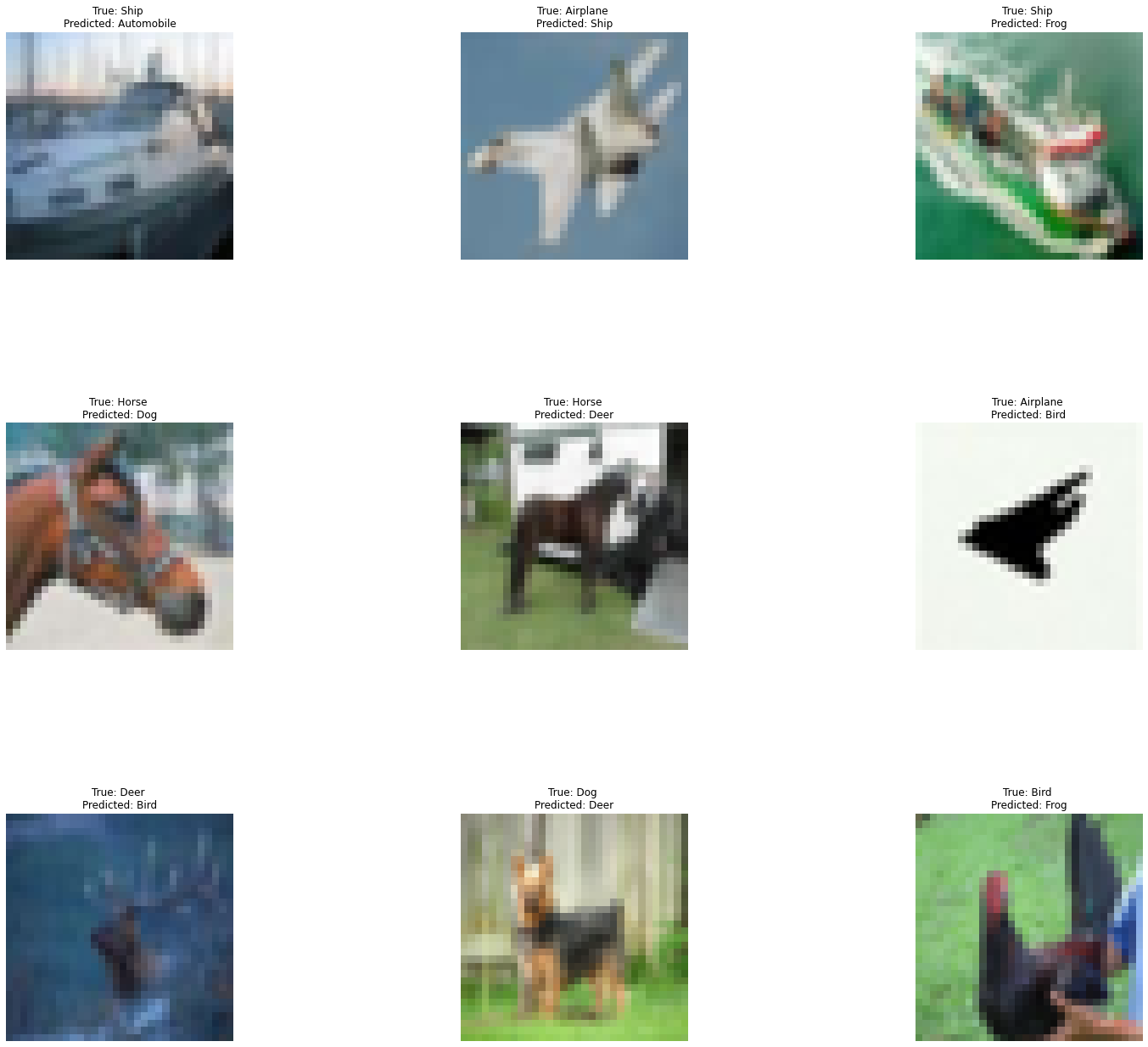
## Creating the CNN

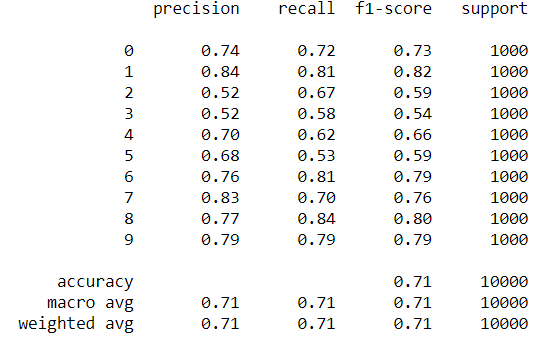
In order to create the convolutional neural network, the keras library was used. A model was first created with 32 convolutional filters with a 3 x 3 size and the relu activation function. This will allow the input image and the output image to have the same shape. A pooling function is then added to reduce the number of parameters that are needed to learn. Another 64 convolutional filter with a relu activation function is utilized, then another 2 x 2 pooling function is used with a drop out of .25. It is then flattened and a dense function that signifies the 10 different labels of classification is added. A final softmax activation function is added in order to help with the classification task.

The model was then compiled with a cross entropy loss function which will is useful for classification tasks. A smaller batch size of 32 was used because the number of images being processed isn’t on a large scale. Different epochs were chosen, but the accuracy did not improve beyond the 5th epoch.

### Results







Based off the testing images, there were some images that were classified correctly and others that were classified incorrectly. The classification report indicates that there was an overall accuracy of .71. However, the classification of images varied between the different labels. Most of the labels received a decent F1 Score between .7 and .8 and above such as: Airplane (0), Automobile (1), Frog (6), Horse (7), Ship (8), Truck (9). The model was able to better classify these different images. The different modes of transportation were easier for the model to distinguish. The remaining labels had a F1 Score below .7: Bird (2), Cat (3), Deer (4), and Dog (5). It was more difficult for the model to distinguish between the different animals. Overall, the model was capable of correctly identifying most of the images, but there are rooms for improvement.

## Part 2 AI Blackjack Strategy

The focus of the second part is developing a strategy for a single player AI to go against the dealer.

### State

* Initial Dealings State
  + Player can buy in to the game with an initial dollar amount.
  + Dealer distributes 2 cards to each player as well to the dealer himself.
    - Dealer will reveal one card while hiding the other card.
  + Dealer checks if he/she has a natural blackjack.
    - If player also has natural blackjack, then player will draw and receive initial bid back.
    - If player does not have a natural blackjack, then player will lose initial bid.
    - Skip to end state.
* Active Playing State
  + Player performs action
    - Player can split and hit until the player decides to stay.
    - Player can hit until the player decides to stay.
    - Player can double and receive card.
* Dealer Playing State
  + Dealer will reveal cards
    - If dealer has reached a card sum of at least 17 points, then round ends.
    - If dealer has less than 17 points, then dealer will hit cards until the dealer has reached a card sum of at least 17 points.
* End State
  + Money is distributed.
    - If the dealer’s hand is greater than the player’s hand.
      * Player loses money on the table.
    - If the dealer’s hand is equal to the player’s hand.
      * Player keeps bid amount for the hand.
    - If the dealer’s hand is less than the player’s hand.
      * Player gains the bid amount for the hand.

### Actions

* Player can choose to Hit.
  + Player will receive one card.
    - If the total sum of cards is greater than 21, then the player busts
    - If the total sum of cards is 21 or less, then the player has the option to hit again or stay.
* Player can choose to double.
  + Player will double the amount that was initially bid, then receive one card.
* Player can choose to split cards if he/she has a matching pair of cards.
  + Player will be required to add another bid for the card that is split.
  + Player will be allowed to choose the Hit action until the same condition is met.
* Player can choose to stay and end his/her turn.

### Rewards and Penalties

* Reward
  + Player receives money for having card total greater than dealer card total.
* Neutral
  + No reward or penalty if player card total is equal to dealer card total.
* Penalty
  + Player will lose bid amount if player card total is greater than 21.
  + Player will lose bid amount if player card total is less than dealer card total.

### Action Rule and Reward and State Distribution

Action Rule

* If the player’s total card sum is 11, then player should double.
* If the dealer’s revealed card is less than a 7, then player should stay.
* If the player’s total card sum is 17 or greater, then player should stay.
* If the player’s hand contains a pair of 8, then player should split.

Reward and State Distribution

* Rewarded for having a larger total card sum of cards than the dealer and receiving money.
* Rewarded for matching the total card sum of cards with the dealer and breaking even.
* Rewarded for dealer busting and receiving money.
* Punished for going over 21 total card sum and losing money.
* Punished for having a smaller card total than dealer and losing money.

Overall, the reward is to accumulate as much money from the plays made by the AI player.

There will be 4 nodes in the input layer to correspond with the 4 states in the game.

There will be a Tanh activation function for the hidden layers. This activation function can correspond with a negative positive, or neutral output which will match the rewards and penalties of the AI player.

There will be 4 nodes as part of the output layer to correspond with the number of actions that are available to the AI player. A softmax activation function will be used for the output layer in order to determine whether the player’s hand will likely be able to win, draw, or lose.

### Strategy

The AI player will check the 2 cards that are given. If the total sum of the cards is 11, then the AI player will always double. The AI player will then check the revealed card of the dealer. If the card that the dealer reveals is less than a 7, then the AI player will stay its hand. If the AI player starts with a hand of 17 or higher, then the player will stay its hand. If the AI player has a pair of 8s as its starting hand, then the player will always split its hands. Outside of these rules, the AI player will learn to hit when it thinks it will be rewarded and when to stay with its hands.