Image Classification of CIFAR 100

Final Milestone

CMPE 257 – Machine Learning

**Team Name -Group 1**

Team Members:

Chaithanya Reddy Bogadi - 013803998

Naga Sindhu Korlapati - 012467832

Nandini Puppala - 013752713

Prathusha Koouri - 013710658

Sindu Ravichandran - 013769824

Venkata Anil Kumar Thota - 012525734

**Best Accuracy obtained with ML Algorithm:**

Milestone 1: CNN with 86.17%

Milestone 2: CNN with 85.33%

Milestone 3: CNN with 77.71%

Milestone 4: CNN with 73.33%

**Google Collaboratory Links:**

**Milestone 3 Collab files:**

<https://colab.research.google.com/drive/1jdKuY8EHxI_Z-ZFokFqYWi1L9e2meK5R#scrollTo=GLIJIi5t1u-t>

* This Collab file contains Traditional Machine learning algorithms for all the 100 trials.

<https://colab.research.google.com/drive/1-u2Hv1FNYbxUHwImlA03oU6G-liLsT3-#scrollTo=Z_UKCxEoi_HY>

* This Collab file contains Traditional Machin learning algorithms for all the 100 trials.

<https://colab.research.google.com/drive/1oBFrvp2LGIW8vUyep9uCXowCTJfkg6MR#scrollTo=n4FnaRipT2EQ>

* This Collab file contains CNN for all the 100 trials by removing 2 subclasses from each superclass.

<https://colab.research.google.com/drive/1A8zA99RDDVro_SraVPsC12s75eo_co1Q#scrollTo=DBjp-PLtTssX>

* This Collab file contains CNN for all the 100 trials by removing 3 subclasses from each superclass.

# **Work Collaboration Table Summary:**

|  |  |  |
| --- | --- | --- |
| Data Cleaning (Milestone-1) | We had to check for the right images, map and separate the images accordingly. | Nandini Puppala, Venkata Anil Kumar Thota, Chaithanya Reddy Bogadi |
| Data Preparation (Milestone-1 and 2) | Normalize the data, checked the distribution and correlations. Separated all the sub-class data. | Naga Sindhu Korlapati, Prathusha Koouri, Sindu Ravichandran |
| Data Preparation (Milestone-1 and 3) | Worked on preparing the test data and train data sets | Nandini Puppala, Venkata Anil Kumar Thota, Chaithanya Reddy Bogadi |
| Code for iterating the test pairs  (Milestone-2) | We created a function to iterate and fit the machine learning algorithms for all the 25 sub-classes. | Nandini Puppala, Venkata Anil Kumar Thota, Chaithanya Reddy Bogadi |
| Code for iterating the test pairs  (Milestone-3 and Bonus) | We modified the function to iterate and fit the machine learning algorithms for all the 100 subsets of the sub-classes. | Naga Sindhu Korlapati, Prathusha Koouri, Sindu Ravichandran |
| List of Algorithms to be used.  (Milestone-1, 2,and 3) | we came up with a list of ML algorithms that work well with image classification data. | Chaithanya Reddy Bogadi, Nandini Puppala, Sindu Ravichandran, Naga Sindhu, Venkata Anil Kumar Thota, Prathusha Koouri |
| Traditional Machine Learning Algorithms (Milestone-1, 2 and 3) | Worked with all the classic classification algorithms, fine-tuned parameters. | Venkata Anil Kumar Thota, Chaithanya Reddy Bogadi, Prathusha Koouri |
| Ensemble Algorithms (Milestone-1,2 and 3) | Tried improving the accuracy with the ensemble methods | Sindu Ravichandran, Nandini Puppala, Naga Sindhu |
| CNN  (Milestone-1,2 , 3 and bonus) | We worked on CNN tuning hyperparameters and regularization parameters to get better results. | Prathusha Koouri,  Nandini Puppala, Sindu Ravichandran, Naga Sindhu, Venkata Anil Kumar Thota, Chaithanya Reddy Bogadi, |
| Report (Milestone-1,2 and 3) | We documented the whole project work in a report. | Chaithanya Reddy Bogadi, Nandini Puppala, Sindu Ravichandran, Naga Sindhu,Venkata Anil Kumar Thota, Prathusha Koouri |
| Powerpoint Presentation  (Milestone-1, 2 and 3) | We worked on PPT together and created the content. | Nandini Puppala, Sindu Ravichandran, Naga Sindhu, Venkata Anil Kumar Thota, Chaithanya Reddy Bogadi, Prathusha Koouri |
|  |  |  |

**Contents**

[**Work Collaboration Table Summary:** 2](#_Toc8476009)

[**Software and libraries used:** 5](#_Toc8476010)

[**Software:** 5](#_Toc8476011)

[**Libraries:** 5](#_Toc8476012)

[**Dataset:** 6](#_Toc8476013)

[**Logistic:** 6](#_Toc8476014)

[**K-NN:** 6](#_Toc8476015)

[**SVM:** 7](#_Toc8476016)

[**Random Forest:** 7](#_Toc8476017)

[**XG Boosting** 8](#_Toc8476018)

[**CNN** 9](#_Toc8476019)

[**Data Preparation Milestone 1:** 10](#_Toc8476020)

[**Data Validation Milestone 1:** 12](#_Toc8476021)

[**Machine Learning Algorithms Summary Milestone 1:** 13](#_Toc8476022)

[**CNN Model for Milestone 1:** 14](#_Toc8476023)

[**Data Preparation Milestone 2:** 16](#_Toc8476025)

[**Data Validation Milestone 2:** 17](#_Toc8476026)

[**Machine Learning Algorithms Summary Milestone 2:** 19](#_Toc8476027)

[**CNN Algorithm Summary Milestone 2 and 3:** 26](#_Toc8476028)

[**CNN Algorithm Summary Milestone 2, 3 and Bonus:** 32](#_Toc8476029)

**Introduction:**

Image classification is a supervised learning problem and is one of the complex tasks for machines, which can be solved using Machine Learning and Deep learning algorithms. Sometimes it is difficult even for humans to differentiate between images of different species. Recent advances in deep learning made tasks such as Image and speech recognition possible. Image classification is performed by taking an image as input and producing an output class as: Binary or Multiclass or their Probabilities. The emphasis is on the summarization of major advanced classification approaches and the techniques which are used for improving classification accuracy.

In this study, for the 1st milestone, we considered two super classes from CIFAR 100 dataset, they are Household Electronics and Household Furniture and our task here is to consider all the 5 subclasses as one class and perform a binary classification on 2 super classes. We applied multiple machine learning algorithms to accomplish this task and finally we were able to achieve excellent results with Convolutional neural networks as they can take the image data as input in 3 dimension and process through the image part by part, using filters which other algorithms cannot.

In 2nd milestone, we had to select one sub-class from each of the assigned super class and use the two selected sub classes as testing data, the remaining data as train set. In this milestone, the emphasis was on the getting the best possible average accuracy for the test set.

In 3nd milestone, we had to select two sub-classes from each of the assigned super class and use the four selected sub classes as testing data, the remaining data as train set with 40% of the subclasses in test data. In this milestone, the emphasis was on the getting the best possible average accuracy for the test set.We worked on a bonus milestone as well. We selected 60% of the data that is 3 subclasses from each superclass as test set. We tried improving the model accuracy with 60% of the sub classes removed from training sets.

# **Software and libraries used:**

## **Software:**

* **Python**: Python comes with a huge amount of inbuilt libraries. Many of the libraries are for Artificial Intelligence and Machine Learning frameworks.
* **Tableau**: It is mainly used for interactive data visualization purpose.
* **Microsoft Excel:** It is spreadsheet that allows user to do calculations, graphing tools, pivot tables.

## **Libraries:**

* **Numpy:** It is mainly used for scientific computing and supports large, multidimensional arrays and matrices.
* **Pandas**: It is one of the most widely used tool in data wrangling/munging.
* **Seaborn:** It is a data visualization library.
* **Matplotlib:** It is used to create 2D plots and graphs using python library.
* **Sklearn:** It is a robust library that provides wide range of supervised and unsupervised learning algorithms.
* **Tensor flow:** It is a fast-numerical computing library that can be used to create deep learning models.
* **Keras:** It is a high-level API built on top of Tensorflow. It is more user-friendly and easier to implement compared to Tensorflow.

# **Dataset:**

CIFAR 100 dataset is available in Keras library. It can be imported from Keras datasets. In this project we worked on CIFAR 100 dataset. This dataset consists of 60000 colored images with dimensions 32\*32\*3, partitioned by 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 super classes. Each image comes consists of two label\_modes:

1. Coarse Labels - Images are partitioned with Superclass label names. (eg: Household electronics, Furniture)
2. Fine Labels - Images are partitioned with Class label names. (eg: lamp, keyboard, chair, etc.)

In this project, data is loaded with fine labels.

For our analysis, we worked on two super classes i.e, Furniture and Electronics.

## 

**Machine Learning algorithms used in the project:**

## **Logistic:**

Logistic regression is the standard solution for binary classification problems. “Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a [logistic function](https://en.wikipedia.org/wiki/Logistic_function)” .Hence we choose logistic as our first algorithm. It is pretty similar to linear algorithm with a key difference that output value modeled to binary values either 0 or 1. It also gives predicted probability as output. These probabilities can be converted into class predictions.

## **K-NN:**

K-NN is a supervised machine learning algorithm. It is one of the popular techniques used while handling large datasets. KNN uses three types of distance functions: Euclidean, Manhattan and Minkowski. In this project, we used Euclidean distance.

KNN classifies a new instance based on the voting criteria.

## **SVM:**

A support vector machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. When labeled training data is given as an input the algorithm outputs an optimal hyperplane. The advantage of SVM is that once a boundary is established most of the training data is redundant. So, we utilized this algorithm for our classification problem.

There are three tuning parameters in SVM classifier. They are:

1)Regularization parameter

2)Kernel and

3)Gamma.

Regularization parameter(C) tells SVM how much we want to avoid misclassifying each training example. For large value of parameter C optimization will choose smaller margin hyperplane,on the other hand for the smaller value of parameter C optimization will choose larger margin hyperplane. Kernel is a method of using a linear classifier to solve a nonlinear problem. These kernel functions can be of different types. For example: linear, nonlinear, polynomial, sigmoid and radial basis function(rbf). gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’.

Using GridsearchCV we got a set of best parameters. The best set of parameters are: C: 1, kernel: rbf and gamma = 0.01.

After working on traditional ML algorithms, we wanted to check if the accuracy improves with ensemble methods which generally reduce bias, variance, noise which will eventually reduce the gap between actual and predicted values.

## **Random Forest:**

Random Forest is one of the popular ensemble technique.It is widely accepted winning algorithm in many data science competitions. Random Forest is a supervised Machine Learning algorithm which is built by picking up features randomly from the data and builts decision trees in making predictions on classification and regression problems.

The main tuning parameters with Random Forest were n\_estimators and max\_depth. Since a forest is built on picking up features from the data. It is important to set how many features, it should pick randomly and how deep it should build the tree. A deep decision trees are prone to overfitting problem. Hence, it is important to look in to that as well.

As we went with more estimators and depth but score is saturated.Hence, we finalized our parameters with no.of estimators as 500 and Max.depth as 300.

**Gradient Boosting:**

There are various boosting techniques in practice which are giving very good results like Gradient boosting, XGboost and Adaboost. In this method more emphasis is on the data which gives wrong predictions in order to improve the accuracy.

Boosting is an ensemble technique which selects the predictors sequentially rather than independently. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. Main logic behind this algorithm is to focus more on residuals and modify the model to predict residuals of the previous models. In the end, all the predictors are combined by giving some weights to each predictor.

Parameter tuning in gradient boosting algorithm can be done in two different levels,

* Tree based parameters:
  + Min\_samples\_split – minimum no.of samples required in a node to split
  + Min\_samples\_leaf – minimum samples required in leaf node
  + Max\_depth - max depth of a tree
  + Max\_features – no of features to be considered while searching for best split.
* Boosting Parameters:
  + Learning\_rate – determines the impact of each tree in the output
  + N\_estimators – no of sequential trees to be modeled
  + Subsample – fraction of observation to be selected for each tree
* Miscellaneous Parameters:
  + Random\_state – random number seed
  + Warm\_start – to add additional trees to the previous fit of the model

## **XG Boosting**

XGboost: Xtreme gradient boosting, actually refers to the engineering goal to push the limit of computational resources for boosted tree algorithms. XGBoost makes the best use of available resources to train the model. Advantages of XGBoost:

* Sparse Aware (handling missing data automatically)
* Block Structure (supports parallelization of tree construction)
* further training (to further boost an already fitted model on new data)
* Out-of-core computing**:** This feature optimizes the available disk space and maximizes its usage when handling huge datasets that do not fit into memory.

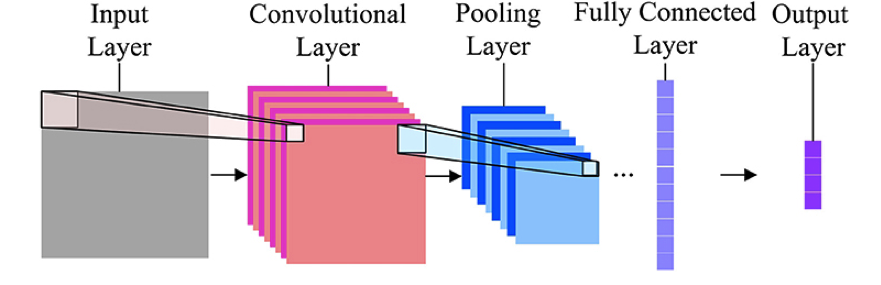
We got a list of best parameters using GridSearchCV. By tuning the following hyperparameters

1. n\_estimators: 50
2. nthread: 4
3. objective: binary-Logistic,
4. learning\_rate: 0.05

## **CNN**

Convolutional Neural Network also known as Convnets or CNN is an artificial neural network, which is good at analyzing spatial and temporal dependencies in an image. It is used in various data analysis and classification problems as well. Wide applications like image recognition, image classification and pattern recognition etc are mostly implemented using CNN.

CNN is made up of several layers that process and transforms an input and produces an output. Technically, CNN model takes each input image and passes through series of convolution layers with filters (kernels), pooling, fully connected layers, and apply activation function to classify an object with probabilistic values between 0 and 1. The complete process is shown in the below diagram.



**Convolution layer:** It is mainly used to extract features in an image like multiple edges and corners etc. In each convolution layer we need to specify the number of filters to have.

**Pooling layer:** It is a kind of down sampling or sub sampling method, which reduces the dimensionality of each map but retains the important information. Methods like max, average and sum pooling are used.

**Fully Connected layer:** In this layer, we flattened our matrix into vectors and feed it into FC layer like neural network, where each and every node is connected to the next layer.

Based on requirements we can apply striding and padding concepts in our model.

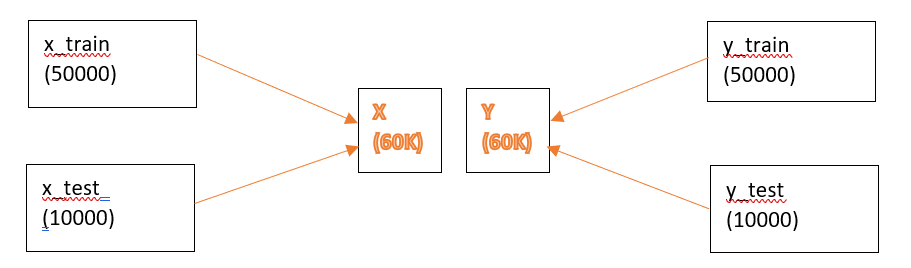
**Hyper Parameters Tuning in CNN**

* **Learning rate:** It controls the update of weight in each optimization algorithm like SGD, Adam, and RMSprop etc.
* **Number of epochs:**It is the number of times entire training set has to pass through neural network.
* **Batch size:** Convnet is sensitive to batch size, and a mini batch ranges from 16 to 128 is a good choice.
* **Activation Function:** It introduces non-linearity to the model. Usually, rectifier works well with convnet. Other alternatives are sigmoid, tanh etc.
* **Number of hidden layers:**It is usually good to add more layers until the test error no longer improves. Less number of layers can also lead to problem of under fitting.
* **Drop out for Regularization:** Dropout is a preferable regularization technique to avoid overfitting in deep neural networks. The method simply drops out units in neural network according to the desired probability. A default value of 0.5 is a good choice to test with.

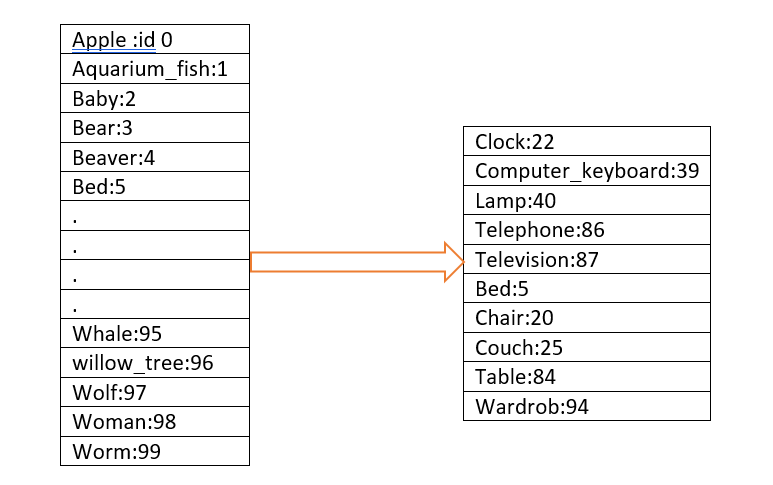
## 

## **Data Preparation Milestone 1:**

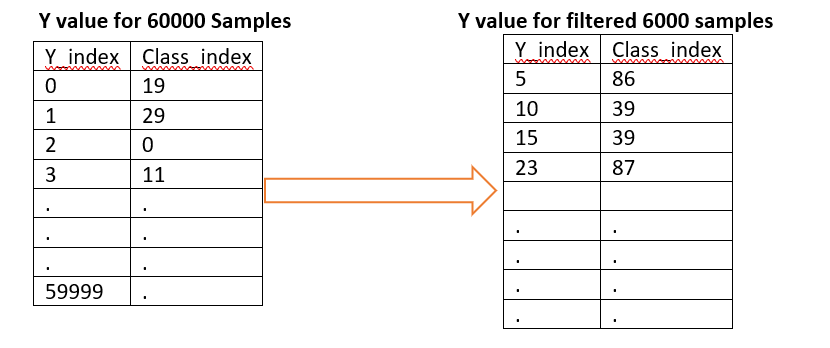
The data from keras is loaded as x\_train,y\_train,x\_test,y\_test. All this dataset are combined to ‘x’ and ‘y’ dataset to filter the required superclass data.



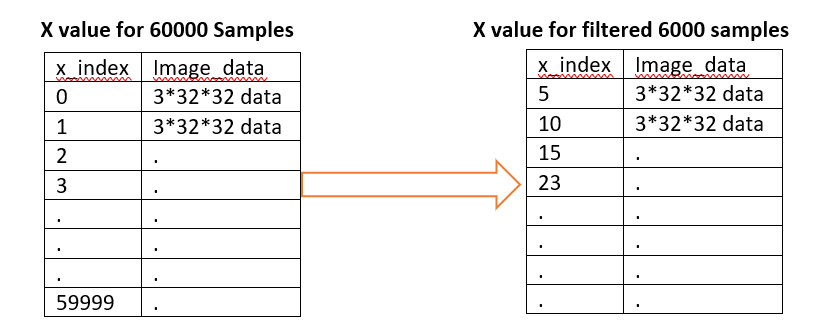
The fine labels of the data are stored in alphabetical order. We created dictionary with all the labels in correct order to filter the class index for the super class ‘Household\_Electronics’ and ‘Household\_Furniture’.



With this class index, we filtered out the index value of the samples in the ‘y’ dataset which belongs to the superclass- Household\_Electronics and Household\_Furniture. 6000 samples are filtered out from the original 60000 samples.

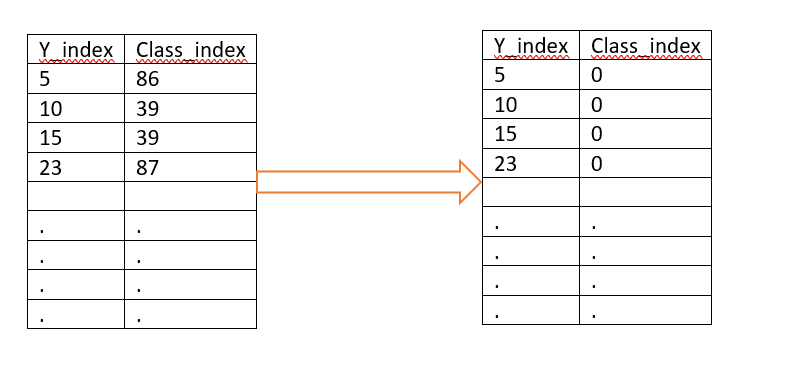


With the filtered ‘y\_index’ value, we were able to filter the ‘x’ dataset also.



The filtered ‘x’ and ‘y’ value contains 6000 images, 600 images for each of the 10 classes. Our main intention is to perform binary classification with the images. Hence, we wanted to update the Target variable ‘y’ to binary format.

* Household\_Electronics : 0 (22,39,40,86,87)
* Household\_Furniture : 1 (5,20,25,84,94)



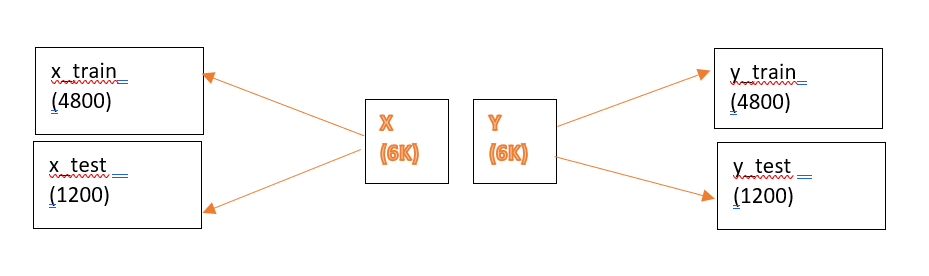
## **Data Validation Milestone 1:**

To validate that the filtered images belongs to the super class Household\_Electronincs and Household\_Furniture, we have displayed 40 images, 4 from each class along with their fine and coarse labels. Below is the grid of images we plotted for validation.



Now the data is ready for applying machine learning algorithms. We use 20% data for testing. Out final train- test dataset shape was,

* X\_train, y\_train : 4800
* X\_test, y\_test: 1200



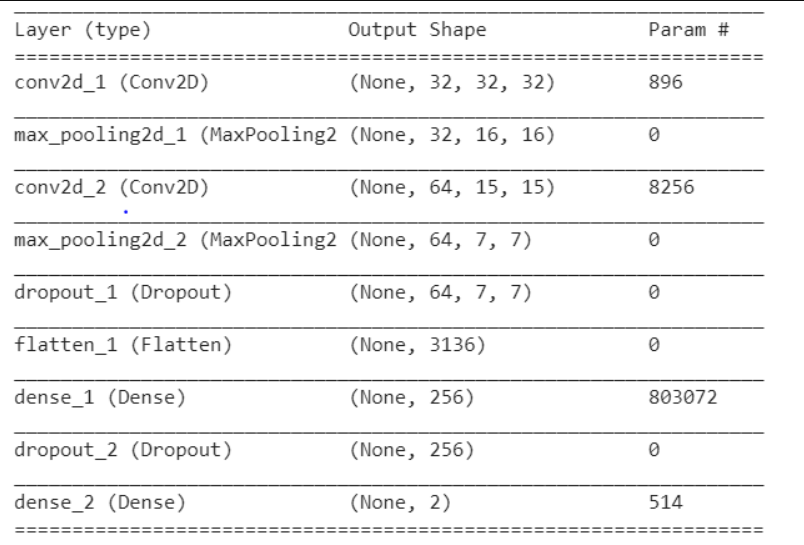
# **Machine Learning Algorithms Summary Milestone 1:**

|  |  |  |
| --- | --- | --- |
| Model | Parameters Tuned |  |
| Support Vector Machine | C =1  Kernel = rbf  Gamma = 0.01 | Default Accuracy: 70.50% |
| Final Accuracy: 79.85% |
| KNN | n\_neighbors = 13 | Default Accuracy: 66.5% |
| Final Accuracy: 73.08% |
| Random Forest | Estimators = 500  Depth = 300 | Default Accuracy: 69.00 % |
| Final Accuracy: 77.30 % |
| Gradient Boosting | Estimators = 300  minsampleleaf = 1  minsample split = 4  max depth = 4  max\_features = auto | Default Accuracy: 71.20 % |
| Final Accuracy:75.25 % |
| XGBoost | n\_estimators = 50  nthread= 4  objective= binary:logistic  learning\_rate= 0.05 | Default Accuracy: 71.20 % |
| Final Accuracy:75.25 % |

## 

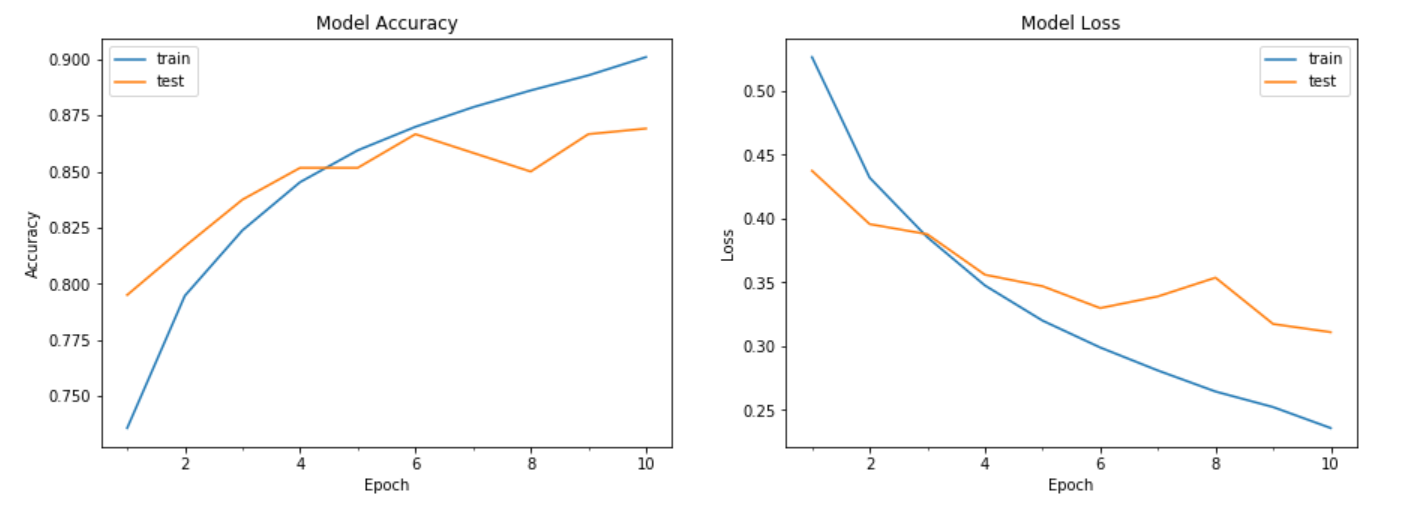
## **CNN Model for Milestone 1:**

## 



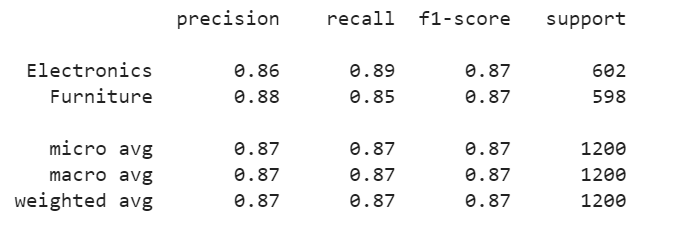
## 

In this CNN model we included one conv layer with 32 filters followed by a max pool layer, another conv layer with 64 filters with drop out (0.5) followed by flattening layer, dense layer with dropout (0.5) followed by dense layer

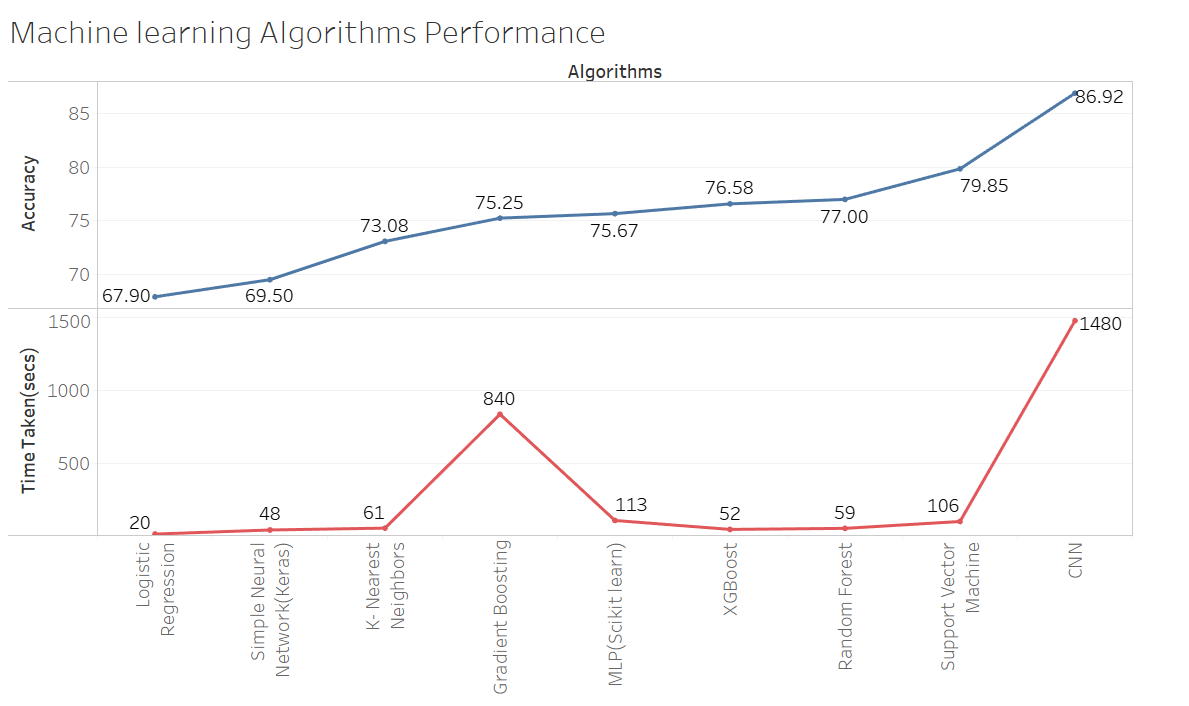


Loss for both test and training data continuously decreased w.r.t to epoch.

Confusion matrix for the best model Milestone 1 - CNN:



Milestone 1:



## Accuracy and Time are the important parameters in building an effective model. Initially we started of working with basic ML model, i.e Logistic Regression which took least amount of time compared to other models and the accuracy was also low. Gradient Boosting took large training time compared to other traditional models, but the accuracy was 75.25%, whereas MLP took least time with best accuracy over the other Machine Learning models. Finally, CNN was top performing model among all the models that we used in our project. Moreover, CNN was the only model which took large training time with high accuracy that is 86.92% in Milestone 1.

## 

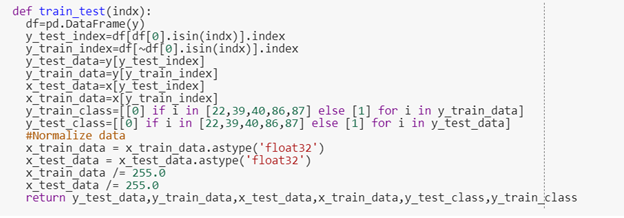
## **Data Preparation Milestone 2:**

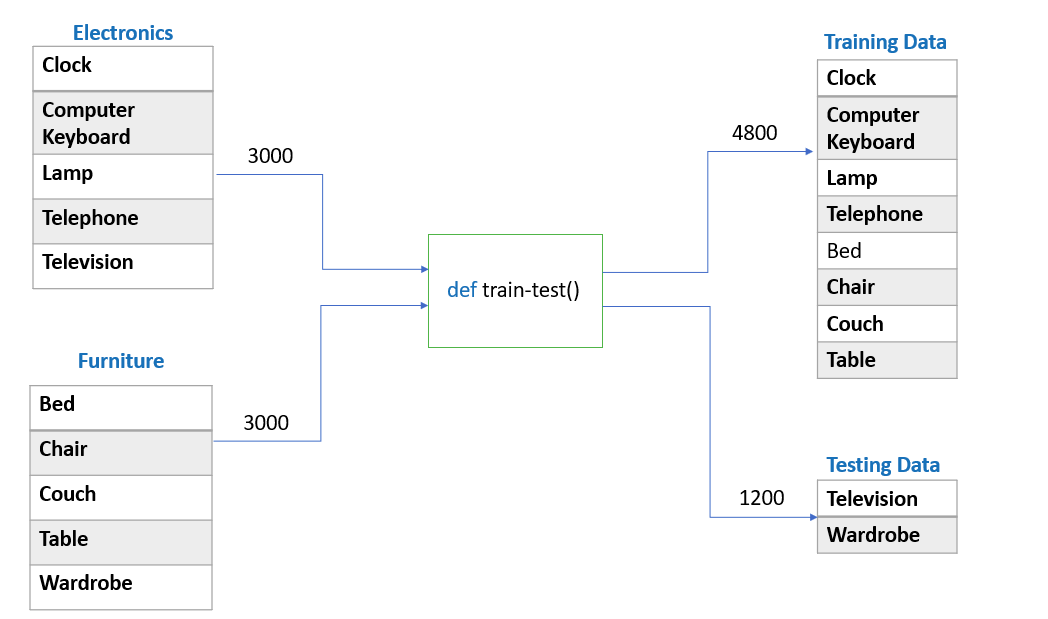
For milestone 2, our requirement is to find the best machine learning algorithm which trains with 8 subclasses and predicts the 2 untrained subclasses. Also, we need to find 2 subclasses that performed best with our algorithms. Hence, we ran our algorithms iteratively for all the 25 subsets. To filter and split the train test data in each iteration, we have created a function named train\_test(). This function takes test class indices as input and returns the train, test data.

Function Name: train\_test(index)

Input: Class index(for test data)

Output: x\_train\_data, x\_test\_data, y\_train\_data, y\_test\_data



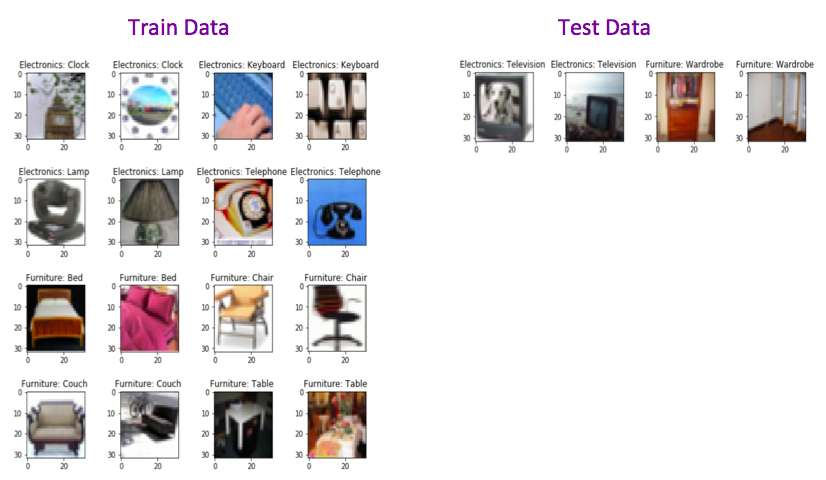


## **Data Validation Milestone 2:**

In order to validate the train and test data, we plotted 2 images randomly from each subclass in one of the test-train set and plotted four images from the corresponding test data.

Train class: Clock, Keyboard, Lamp, telephone, bed chair, couch and table.

Test class: Television and Wardrobe.



**Best model in Milestone 2:**

CNN is the best model for milestone 2.

A close up of a map

Description generated with very high confidence

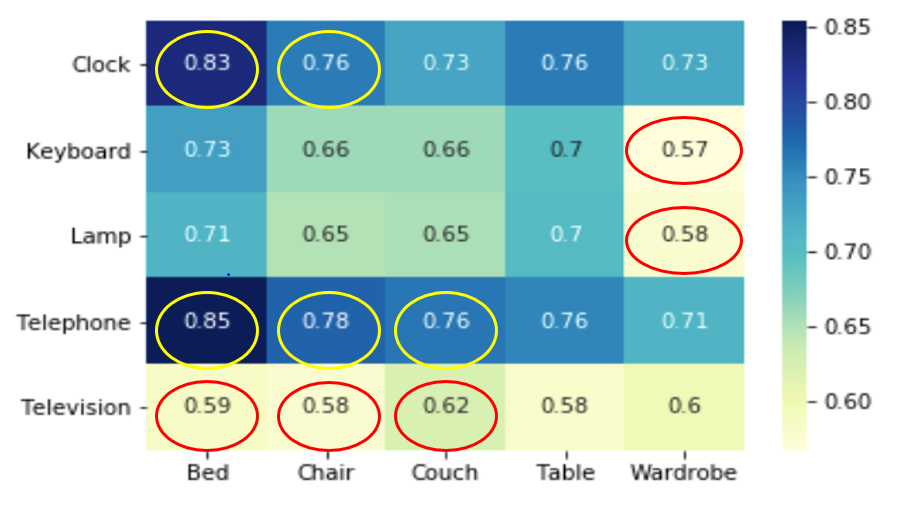
The above diagram depicts the loss function and accuracy curves with respect to the epochs. The loss for both test and training data fairly decreasing w.r.t to epoch.

# **Machine Learning Algorithms Summary Milestone 2:**

Below table shows descriptive statistics for all models used for milestone2.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Min | Max | Average | Median | S.D | 25 Percentile | 75 Percentile |
| Logistic | 50.00 | 64.25 | 58.58 | 60.16 | 0.04 | 56.16 | 61.41 |
| KNN | 41.41 | 65.91 | 56.96 | 57.91 | 0.061 | 54.75 | 61.41 |
| Random Forest | 51.24 | 64.25 | 58.58 | 60.16 | 0.048 | 59.58 | 66.16 |
| XGBoost | 53.42 | 70.25 | 63.05 | 64.67 | 0.057 | 59.92 | 66.25 |
| Gradient Boosting | 55.00 | 71.33 | 63.50 | 64.66 | 0.042 | 61.33 | 66.16 |
| SVM | 51.08 | 71.92 | 64.20 | 66.00 | 0.059 | 63.08 | 68.66 |
| CNN | 56.75 | 85.33 | 69.01 | 69.83 | 0.089 | 62.33 | 75.50 |

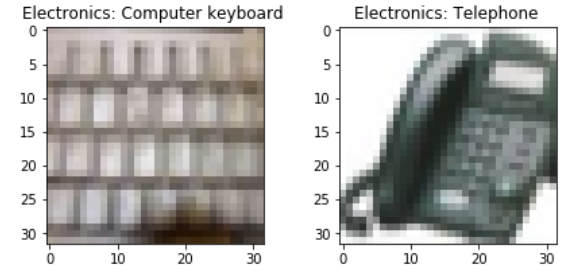
**Heat map for Milestone 2 Final Model:**

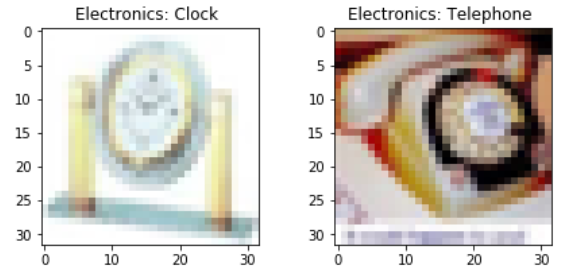


**Top-5 best subsets in MileStone 2:**

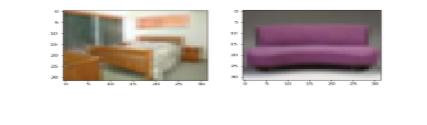
|  |  |  |
| --- | --- | --- |
| **Test SubClass: (Electronics)** | **Test SubClass: (Furniture)** | **Model Score** |
| Telephone | bed | 0.85 |
| Clock | bed | 0.83 |
| Telephone | Chair | 0.78 |
| Telephone | Couch | 0.76 |
| Clock | Chair | 0.76 |

Having Keyboard/Clock in the train set our model is able to predict telephone in test data, since they have similar characteristics as the keypad. When telephone is in train data, model is able to predict clock because some telephone images have round structured keypad which looks like clock.





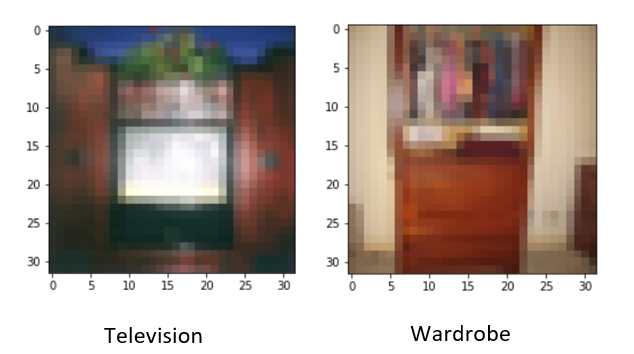
Presence of Couch in train data is helping the model in predicting the bed. Due to similar shape and structure, bed in test data performed good.



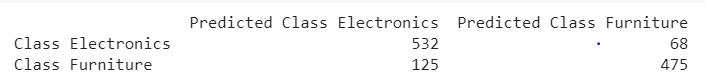
Top 5 Least subsets for Milestone 2

|  |  |  |
| --- | --- | --- |
| **Test SubClass: (Electronics)** | **Test SubClass: (Furniture)** | **Model Score** |
| Keyboard | Wardrobe | 0.57 |
| Lamp | Wardrobe | 0.58 |
| Television | Chair | 0.58 |
| Television | Table | 0.58 |
| Television | Bed | 0.59 |

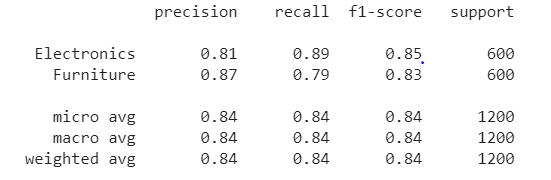
There are very less images with shapes that are similar to Television could be one reason, why model performed worst in predicting the television and ,there are some images like wardrobe which resemble a television.



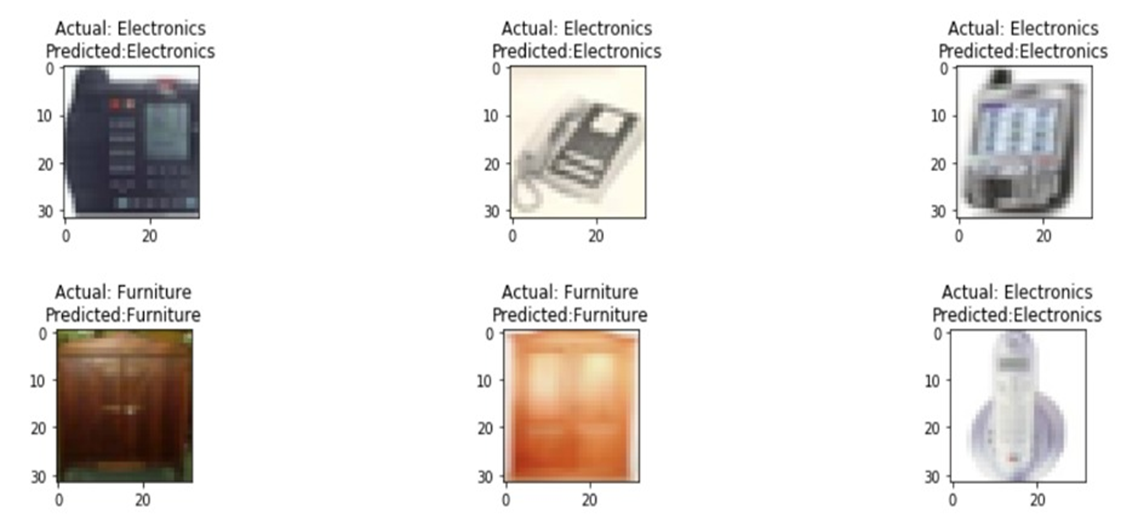
**Confusion Matrix for Milestone 2:**

****

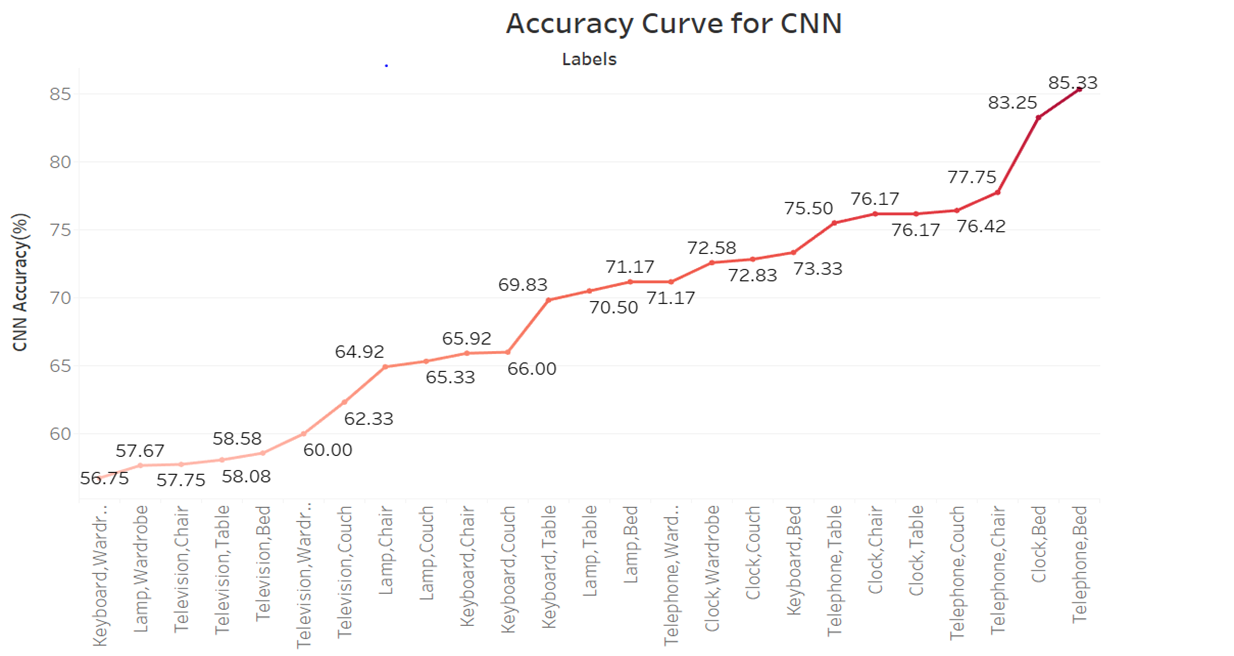
**Classification Report for Milestone 2:**

****

**Predicted Images for Milestone 2:**

****

**Accuracy Curve for CNN:**

****

We built the baseline architecture for CNN using the test labels that performed well with the traditional Machine Learning. With various combinations on all the five subclasses from each super class, we ran 25 test-train class pairs on the CNN Model. keyboard-wardrobe pair in the test class, has given low accuracy score i.e 56.75%. Using Telephone-bed combination in the test class, gave the best score i.e 85.33%.

Comparing and contrasting the performance of various traditional ML algorithms with Deep learning neural networks with the train-test set combinations in milestone 2, CNN performed better in this case as well.

**Milestone 3**

**Data Preparation:**

For milestone 3, our requirement is to find the best machine learning algorithm which trains with only 6 subclasses and predicts the 4 untrained subclasses. Also, we need to find 4 subclasses that performed best with our algorithms. Here, we have run our algorithms iteratively for all 100 subclass pairs. To filter and split the train test data in each iteration, we have used the function train\_test(), that we created for Milestone 2. This function takes test class indices as input and returns the train, test data.

Function Name: train\_test(index)

Input: Class index(for test data)

Output: x\_train\_data, x\_test\_data, y\_train\_data, y\_test\_data

**A screenshot of a cell phone

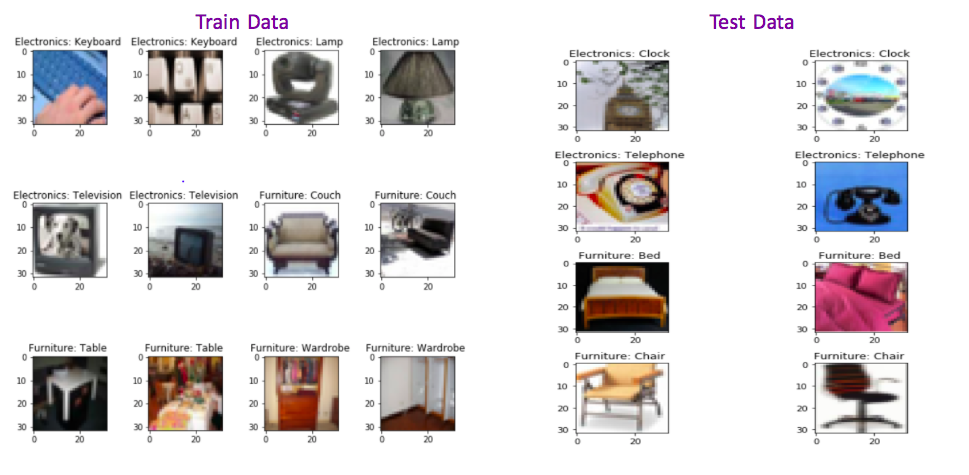
Description generated with high confidence**

**Validation**

In order to validate the train and test data, we plotted 2 images randomly for each subclass from the test-train set, which gave the best accuracy and plotted four images from the corresponding test data.

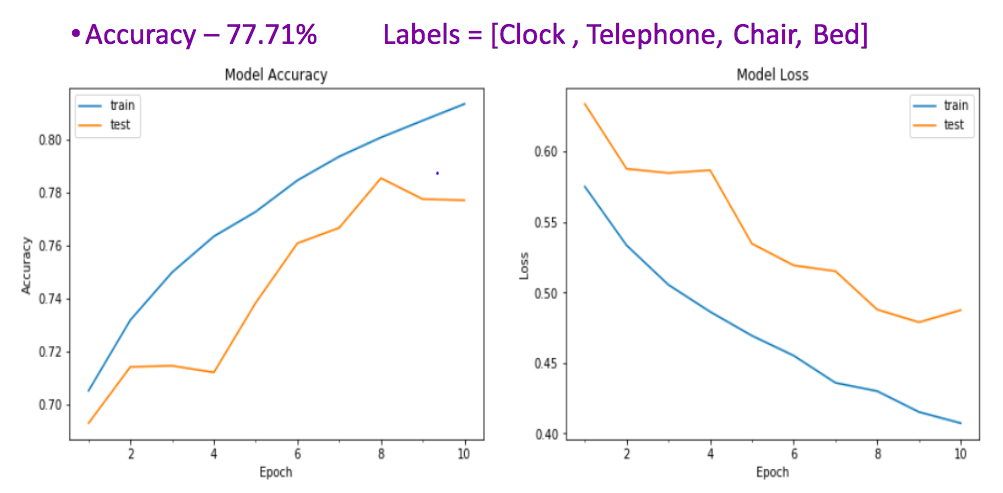
Train class: Keyboard, Lamp, Television,Couch ,Table and Wardrobe.

Test class: Clock,Telephone, Bed and Chair.

****

**Results and observation**

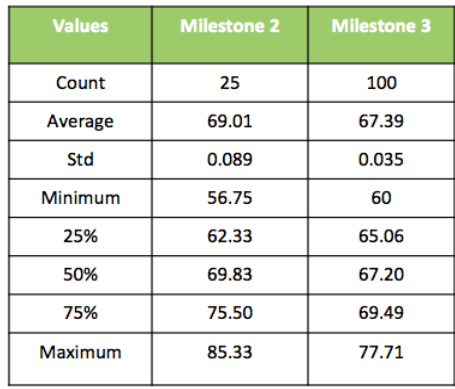
Below is our best model in Milestone 3 with accuracy 77.71%.

****

The above diagram depicts the loss function and accuracy curves with respect to the epochs. The loss for both test and training data fairly decreasing w.r.t to epoch.

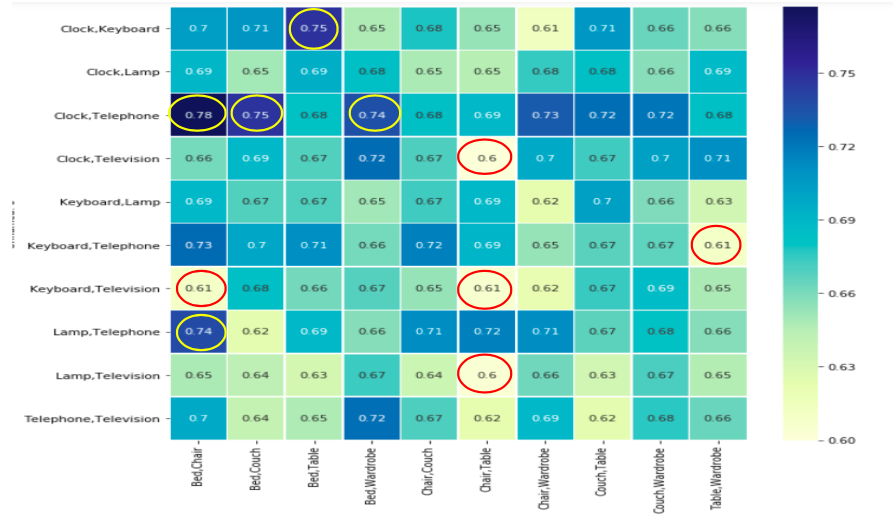
# **CNN Algorithm Summary Milestone 2 and 3:**

Below table shows descriptive statistics for our best model CNN used in milestone 2 and 3.

****

**Heat Map**

Below is Heatmap for our best model in Milestone 3.

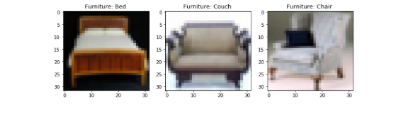
****

**Top- 5 best subsets in Milestone 3**

|  |  |  |
| --- | --- | --- |
| Test SubClass: (Electronics) | Test SubClass: (Furniture) | Model Score |
| Clock, Telephone | Bed, Chair | 0.78 |
| Clock, Telephone | Bed, Couch | 0.75 |
| Clock,Keyboard | Bed,Table | 0.75 |
| Clock,Telephone | Bed,Wardrobe | 0.74 |
| Lamp, Telephone | Bed, Chair | 0.74 |

When we have Keyboard in train class model is able to identify (telephone, clock) in test class. Since the features like keypad are contributing for model in detecting the unseen test image, that is, telephone.

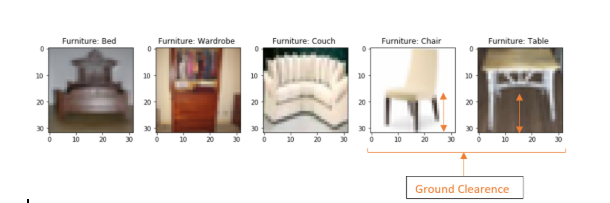
When Couch, Chair are in train data, due to similar shape and structure, our CNN model is able to predict bed in test data.



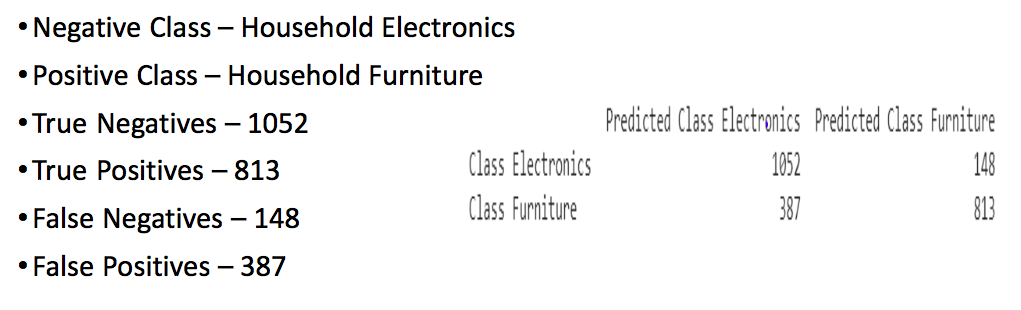
**Top -5 Least score subsets in Milestone 3**

|  |  |  |
| --- | --- | --- |
| Test SubClass: (Electronics) | Test SubClass: (Furniture) | Model Score |
| Lamp, Television | Chair,Table | 0.60 |
| Clock,Television | Chair,Table | 0.60 |
| Keyboard,Television | Bed,Chair | 0.61 |
| Keyboard,Television | Chair,Table | 0.61 |
| Keyboard,Telephone | Table,Wardrobe | 0.61 |

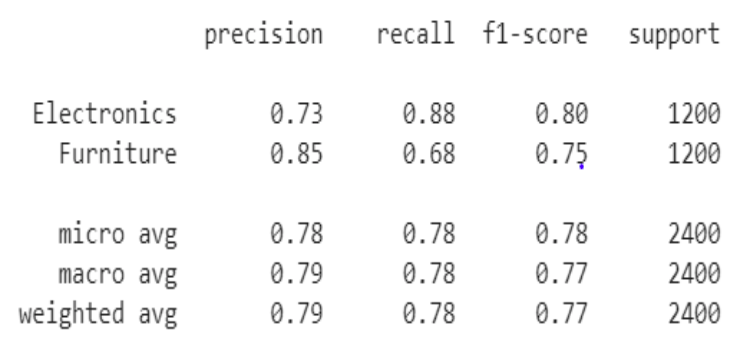
In Furniture Test Class, ( Chair, Table ) have given least score with our model. There is a unique feature which is differentiating other images i.e., ground clearance. Since Table and Chair has high ground clearance, which could be a differentiating factor. As training images don’t have the same characteristics with Chair and Table in test class.This could be one of the reasons for this subset to have least score.



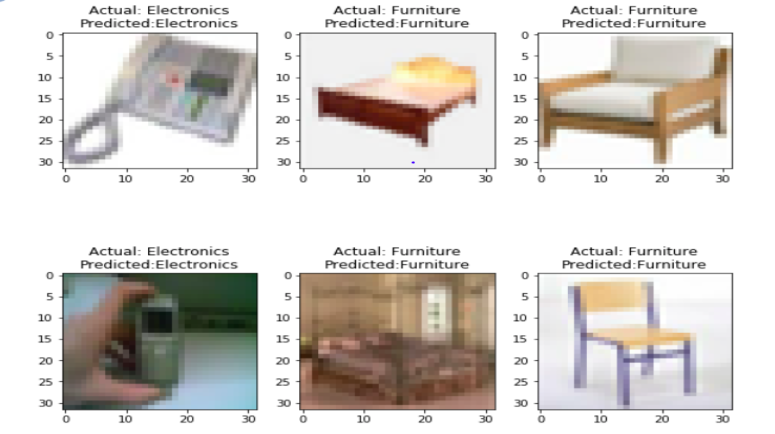
**Confusion Matrix for Milestone 3:**

****

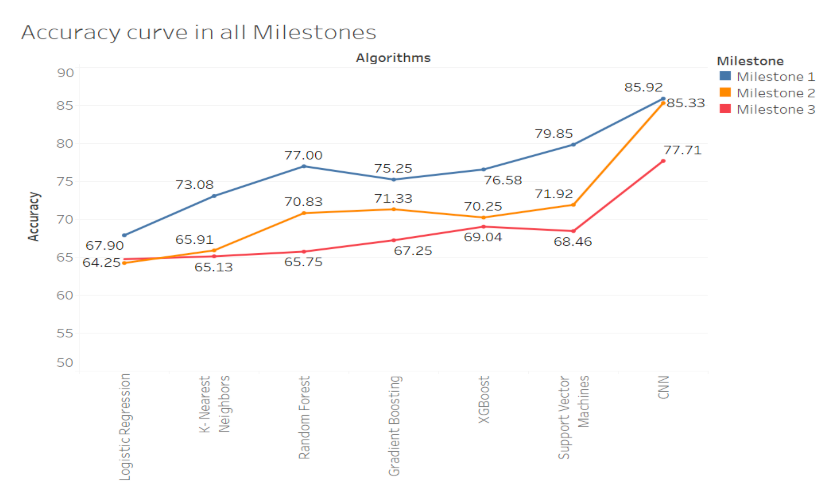
**Classification Report for Milestone 3:**

****

**Predicted Images for Milestone 3:**

****

**Accuracy Curves in all Milestones:**

****

From the graph we can observe that in all three milestones, the CNN model has achieved the highest accuracy and our second best model is Support vector machines. On the other side, for all three instances, the Logistic regression model has less accuracy. From this we can infer that simple Machine learning techniques are not suitable for doing image processing tasks.

**Bonus:**

**Data Preparation:**

For milestone 4, our requirement is to find the best machine learning algorithm which trains with only 4 subclasses and predicts the 6 untrained subclasses. Also, we need to find 6 subclasses that performed best with our algorithms. Here, we have run our algorithms iteratively for all 100 subclass pairs. To filter and split the train test data in each iteration, we have used the function train\_test(), that we created for Milestone 2. This function takes train class indices as input and returns the train, test data.

Function Name: train\_test(index)

Input: Class index(for train data)

Output: x\_train\_data, x\_test\_data, y\_train\_data, y\_test\_data

A screenshot of a cell phone

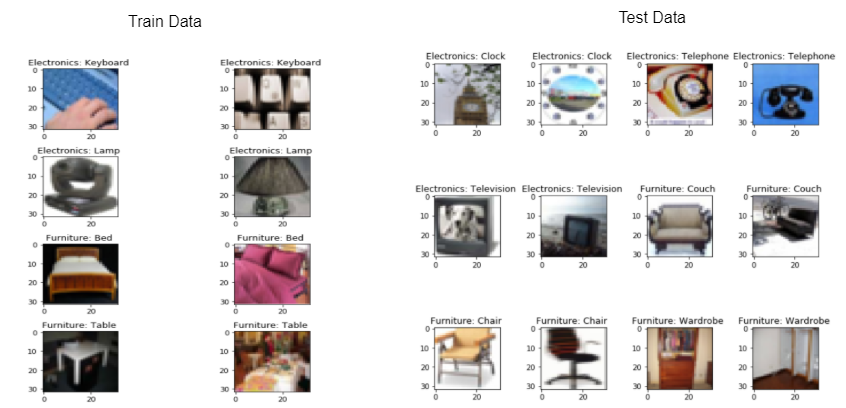
Description generated with high confidence

**Validation**

In order to validate the train and test data, we plotted 2 images randomly for each subclass from the test-train set, which gave the best accuracy and plotted four images from the corresponding test data.

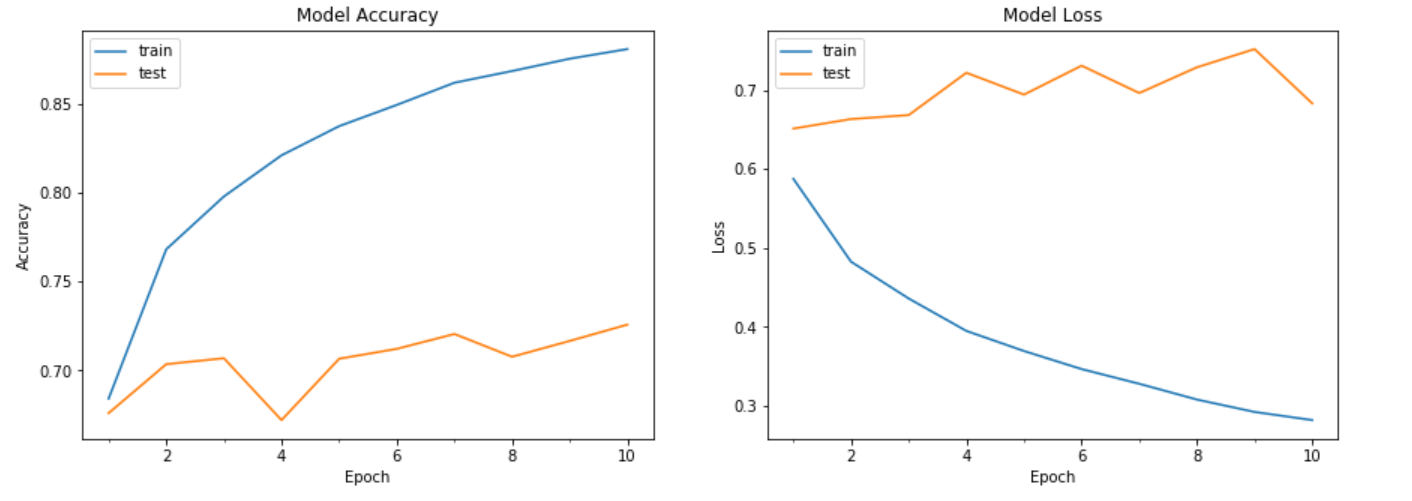
Train class: Keyboard, Lamp,Table and Bed.

Test class: Clock,Telephone, Television, Couch, Wardrobe and Chair.



**Results and observation**

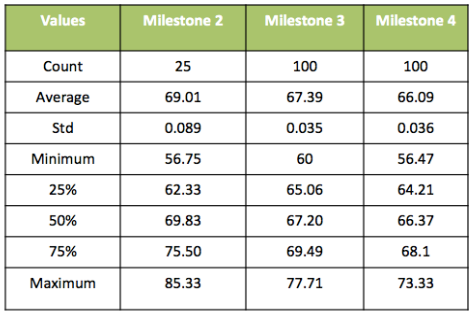
Below is our best model in Bonus Milestone with accuracy 73.33%.



The above diagram depicts the loss function and accuracy curves with respect to the epochs. The loss for both test and training data fairly decreasing w.r.t to epoch.

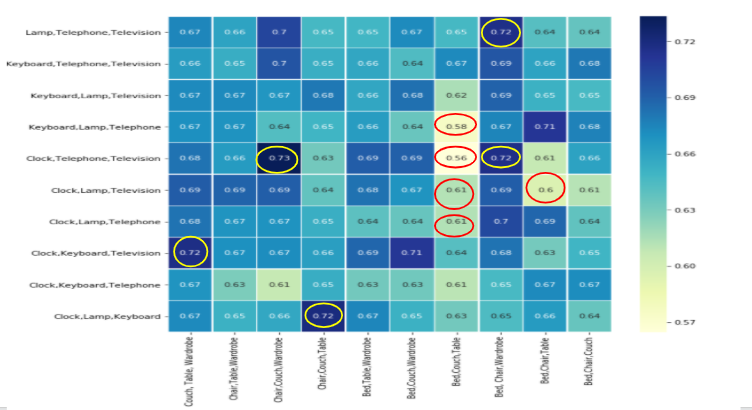
# **CNN Algorithm Summary Milestone 2, 3 and Bonus:**

Below table shows descriptive statistics for our best model CNN used in milestone 2, 3 and Bonus.



**Heat Map**

Below is Heatmap for our best model in Bonus Milestone.



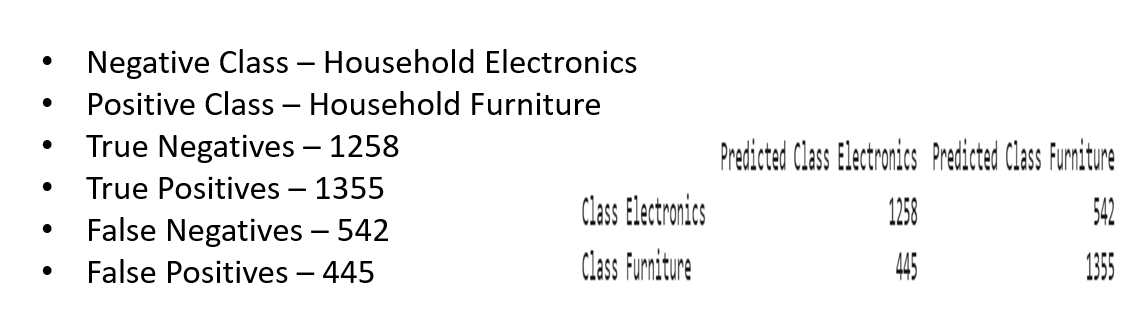
**Top- 5 best subsets in Bonus Milestone**

|  |  |  |
| --- | --- | --- |
| Test SubClass: (Electronics) | Test SubClass: (Furniture) | Model Score |
| Clock, Telephone, Television | Chair, Wardrobe, Couch | 0.73 |
| Clock, Keyboard, Television | Couch, Table, Wardrobe | 0.72 |
| Clock,Keyboard, Lamp | Chair, Couch, Table | 0.72 |
| Clock, Telephone, Television | Bed,Chair, Wardrobe | 0.72 |
| Lamp, Telephone, Television | Bed, Chair, Wardrobe | 0.72 |

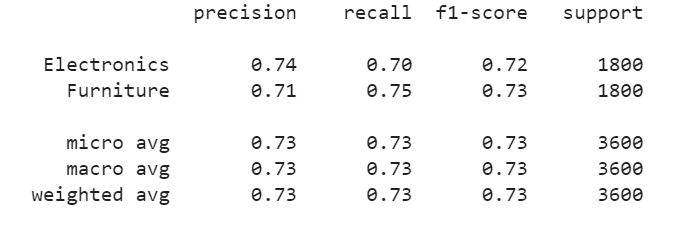
**Top -5 Least score subsets in Bonus Milestone**

|  |  |  |
| --- | --- | --- |
| Test SubClass: (Electronics) | Test SubClass: (Furniture) | Model Score |
| Clock, Lamp, Telephone | Bed, Couch, Table | 0.61 |
| Clock,Television, Lamp | Bed, Couch, Table | 0.61 |
| Clock, telephone, Television | Bed, Couch, Table | 0.56 |
| Keyboard, Lamp, Telephone | Bed, Couch, Table | 0.58 |
| Clock,Television, Lamp | Bed, Chair, Table | 0.60 |

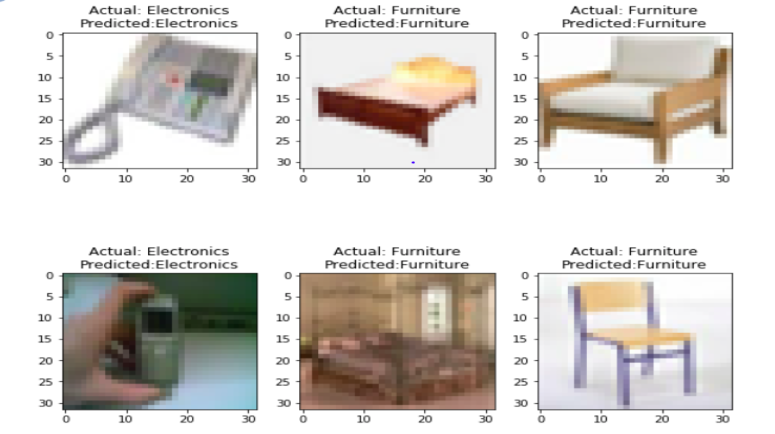
**Confusion Matrix for Bonus Milestone:**



**Classification Report for Bonus Milestone:**

****

**Predicted Images for Bonus Milestone:**

****

**Lessons Learnt:**

1. We could learn to manipulate and work with the 3D image data (RGB) and multidimensional data,
2. Working for the first time with Image augmentation,we observed a lot of change in accuracy by implementing image augmentation like zooming , horizontal flip , vertical flip changing.
3. We tried converting images to grayscale, but this did not increase accuracy.
4. Using gridsearchCV for tuning hyperparameters for traditional machine learning models has helped a lot to find the best parameters and best model.
5. We used kernel and bias initialization to reset the initial weights with random values.
6. We saved the model to retrieve back the model without losing previous tied weights.
7. We used GPU and TPU for processing, which reduced processing by 40 times.
8. Our model is very robust as we have taken care of overfitting by using regularization(drop out layers) in our 1st milestone itself, which helped us have the same model for all the milestones and gave good results.

**Conclusion and Future work**:

With the specific work we did in this project for all the milestones, comparing and contrasting the performance of various traditional ML algorithms with Deep learning neural networks with this dataset, we found that implementing a CNN for identification of label type of household dataset images gave promising results with an accuracy of 77.71% even after we removed 40% of the sub-classes for training set, but the computational performance is a little low. However, we believe that with some further improvements of parameters we could improve the accuracy sufficiently. We would recommend further investigations into the use of Convolutional neural networks, Resnet and VGG net for the Image classification project, as we believe that it has great potential, specially in dealing with image recognition problems.

**References:**

* <https://medium.com/datadriveninvestor/an-introduction-to-grid-search-ff57adcc0998>
* <https://medium.com/@kesarimohan87/model-selection-using-cross-validation-and-gridsearchcv-8756aac1e9d7>
* <https://github.com/tatsuyah/CNN-Image-Classifier>
* <https://github.com/sebsquire/Dogs-and-cats-image-classification-CNN>
* <https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8>
* <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>
* <https://towardsdatascience.com/a-walkthrough-of-convolutional-neural-network-7f474f91d7bd>
* <https://skymind.ai/wiki/multilayer-perceptron>
* <https://medium.com/@mohtedibf/in-depth-parameter-tuning-for-knn-4c0de485baf6>
* <https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python>
* <https://keras.io/initializers/>
* <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>