Assignment – 3

1.1 Describe the dataset, the format selected and why you selected that format?

SVHN dataset was obtained from large street view images and from these images house number patches are selected and numbers were detected using a dedicated sliding window house number detector [4].

The dataset is available in two formats full numbers and cropped digits. In cropped digits images have been resized to 32-by-32 pixels which makes it very easy to work on dataset as all images are of similar size.

For this study, I have used cropped digits dataset because first as it is resized to 32-by-32 pixels it makes very easy to use images of same sizes. Second, as all the images are cropped so max, we can get 10 digits and can-do label binarizer as we will see later in this study.

1.2 Brief summary of approaches used in literature?

In literature [4], researchers have used two different feature learning algorithms stacked sparse auto-encoders and the k-means based system in [2].

For sparse auto encoders researchers have adopted the greedy layer wise stacking approach. In each layer of stacked encoder, they trained an auto-encoder with k hidden units and using back propagation to minimize squared reconstruction error. After this learning procedure, the decoding layer is discarded, and the encoder is used as a nonlinear function that maps its inputs to a K-dimensional feature vector. Once feature representation is done, they added a final softmax classification layer that is trained using maximum likelihood. And the whole network is tuned using back propagation.

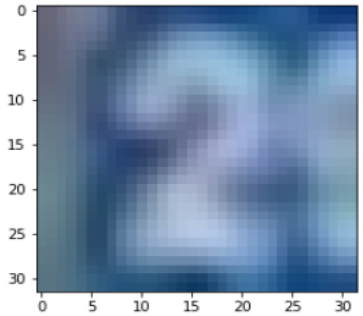
They have also used the K-means-based feature learning system described by [2]. This system works by first extracting a set of 8-by-8-pixel grayscale patches from training images and then applying a variant of K-means clustering to learn a large bank of K linear filters, where K is the number of centroids produced by K-means.

1.3 Describe any pre-processing used and why?

We have shaped data and set axis in addition to that we have done one hot encoding i.e for all categories we represented them into 0's and 1's. for example in screenshot below label for figure 1 without encoding is 2 and for figure 2 encoding is [0 1 0 0 0 0 0 0 0 0].

Figure 1

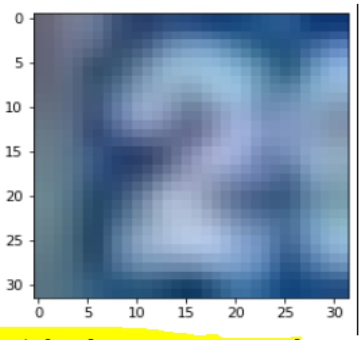
```
plot.imshow(train_imgs[10522])
plot.show()
print('Label: ', train_labels[10522])
```




```
Label:  [2]
```

Figure 2

```
plot.imshow(train_imgs[10522])
plot.show()
print('Label: ', train_labels[10522])
```




```
Label:  [0 1 0 0 0 0 0 0 0 0]
```

Second, we did data augmentation as we know in neural networks the more, we have data to train on our model the better it is for our model to generalize so we use methods like rotating images, zooming doing height shift and expanded our dataset so that we will have more data for training purposes.

```
[ ]  #data expansion
     train_datagen = ImageDataGenerator(rotation_range=8,
                                        zoom_range=[0.95, 1.05],
                                        height_shift_range=0.10,
                                        shear_range=0.15)
```

2.1 Brief description how CNN works?

A classic CNN have following layers:

- Convolutional layer
- Activation layer
- Pooling layer
- Fully connected layer

1. Convolutional Layer

   We have a filter which lays over some of the pixels of the input image depending on the dimensions of the kernel size. The kernel actually slides over the input and keep multiplying the values in the filter and hence sliding the kernel and generates a feature map.

2. Activation Layer

   So as convolution is a linear operation and images are nonl-inear so we use activation function such as Sigmoid, Tanh and ReLU.

3. Pooling Layer

   The pooling layer replaces the output of the network at certain locations by deriving summary stat from nearby outputs. This help in reducing size of representation which decreases the required amount of computation and weights.

4. Fully connected layer

   Fully connected layers involve weights, biases, and neurons. Connects neurons in one layer to neurons in second layer.

2.2 Describe how you split the dataset?

We divided train set into validation set also. So, we kept 85% of data as train set and 15% of data for validation test. And for testing we have already separate data in SVHN dataset.

```
X_train, X_val, y_train, y_val = train_test_split(train_imgs, train_labels, test_size=0.15, random_state=22)
```

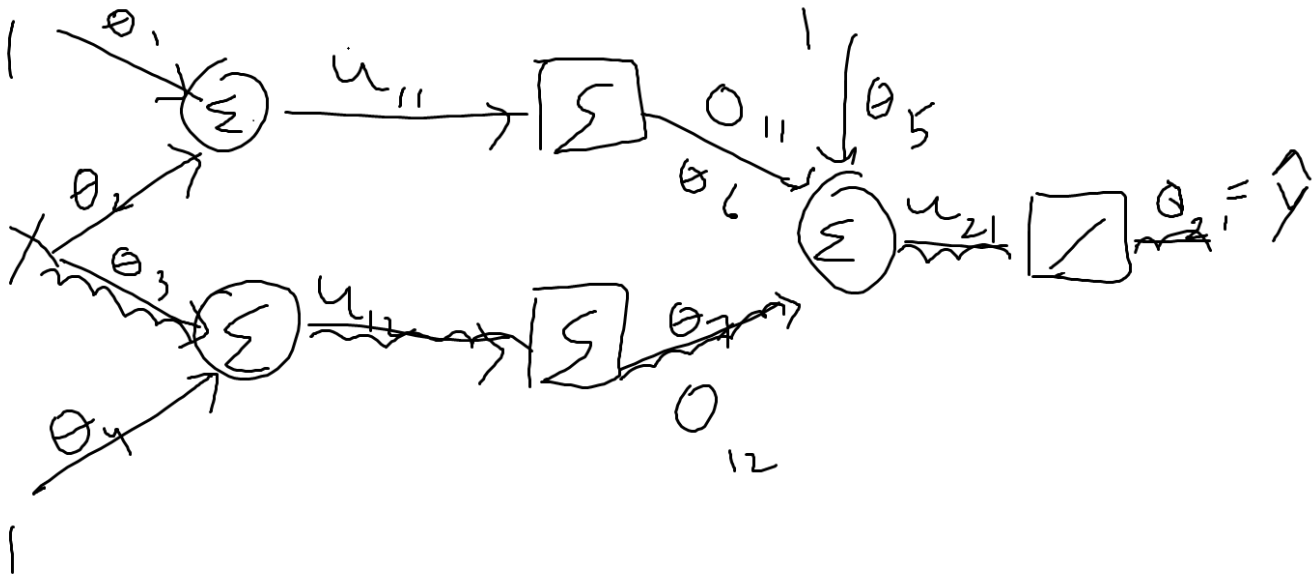
2.3 Select an evaluation metric and explain why you select that?

For evaluation we are using a very simple evaluation metric called accuracy which gives us how many datapoints were predicted correctly. Why we are using accuracy because in our dataset we have cropped images and those images have 10 classes. And for each class we will be having some data and it would not be like 98% of data will be one class and rest from others. So having data for each class will not give false accuracy results and we can rely on accuracy as n evaluation metric for our CNN models.

2.4 Describe the obtained results.

2.4.1 Model 1 with Sigmoid activation function

1. Model is highly biased (simple model) because it did not fitted data well therefore, we saw underfitting.
2. As we used sigmoid as activation function, we faced vanishing gradient problem as explained below.



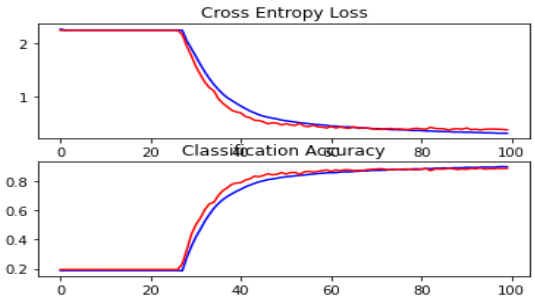
Equation for back propagation for above model.

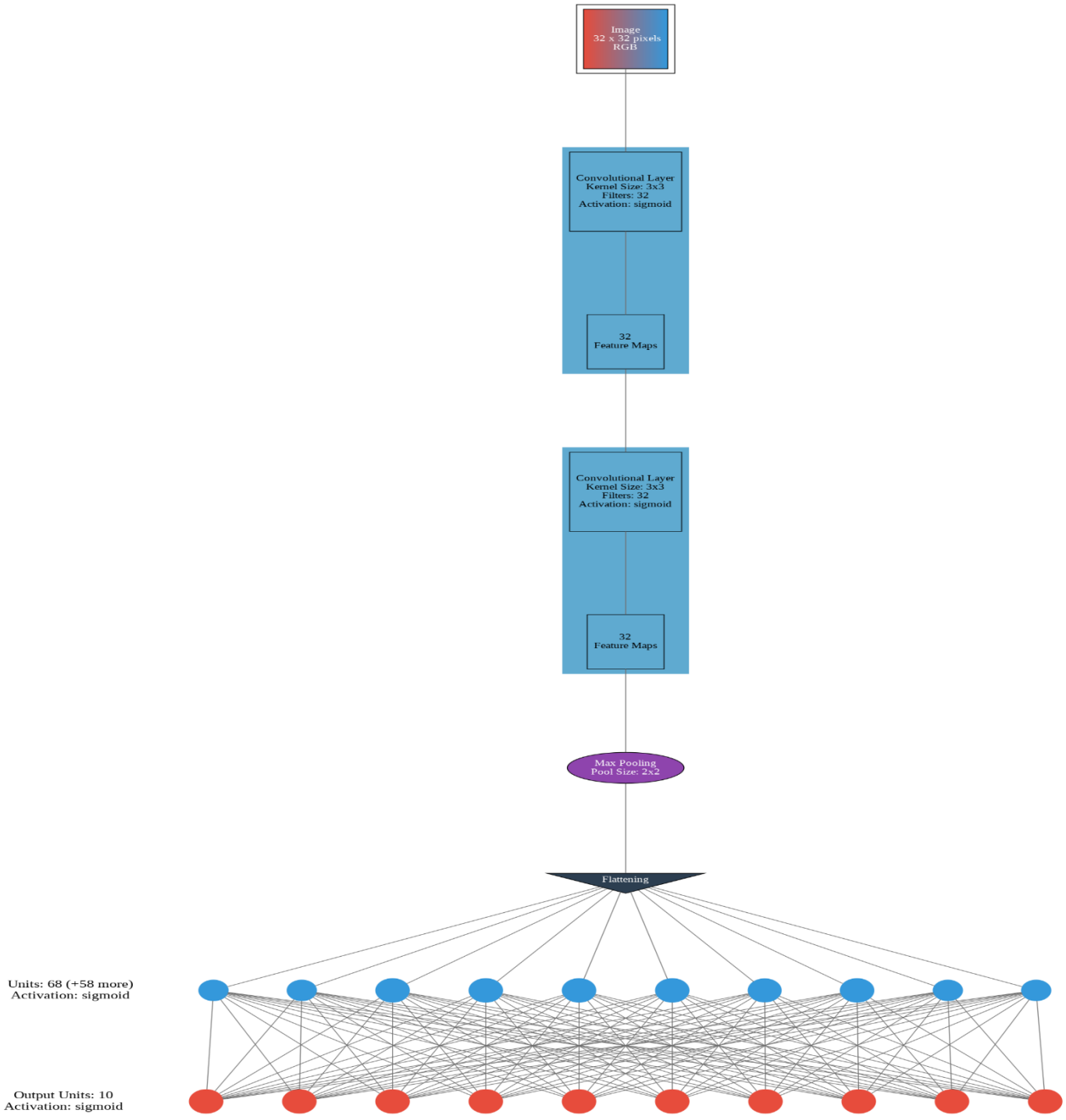
$$\theta_3^i = \theta_3^{i-1} + n(y_i - \hat{y}_i)\theta_7^{i-1}(O_{12}^i(1 - O_{12}^i))X_i$$

so, as we used sigmoid activation function in our Model 1 and one problem that comes with sigmoid is while doing back propagation to adjust our weights by minimizing the error function. For example, in above representation if you see we are trying to find derivative on $\frac{\partial \hat{y}}{\partial \theta_3}$.

Vanishing Gradient Problem

So, the problem that comes is we have to multiply $O_{12}^i(1 - O_{12}^i)$ which are nothing but output from sigmoid which will be between 0 and 1. So if the unit is vey active or not active this product will be 0 or close to 0. So, if gradient becomes 0 the weights don't change. To fix this problem we will use ReLU as the activation function and because with ReLU value is not between 0 and 1 but ReLU can have max value greater than 1 ie between 0 to infinity which solves our problem of making product near or equal to 0.



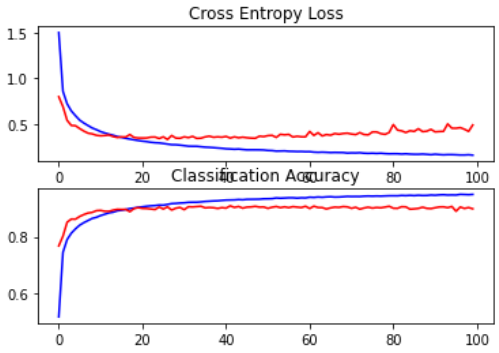


2.4.2 Model 2 with ReLU activations.

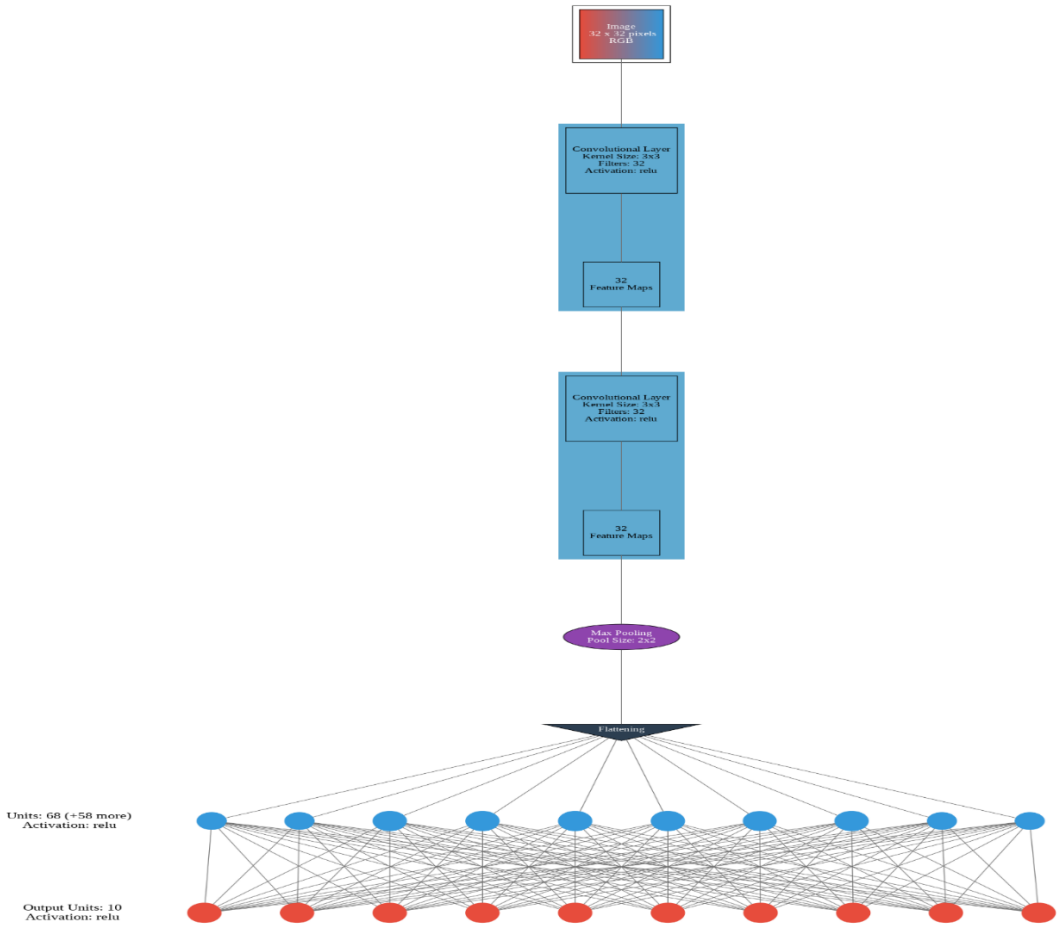
- So, in this model we kept everything same as Model 1 but only difference is instead of using sigmoid as the activation functions, we used ReLU activations to overcome the vanishing gradient problem

ad we see some surprising results. We got accuracy of just above 95% on our training set which is very good as compared to what we got in Model 1.

- Hence, we can say that with use of ReLU activation we have overcome the problem of vanishing gradients as we saw in our Model 1.
- Another surprising thing that we noticed in Model two is that, with training set we got accuracy around 95% but our model did not perform well on testing set and it was just below 89%. Which show that our CNN is having high variance and model is not generalized at all due to which we saw overfitting. That is our model is overfitting the training set very much because of that it failed on test set.

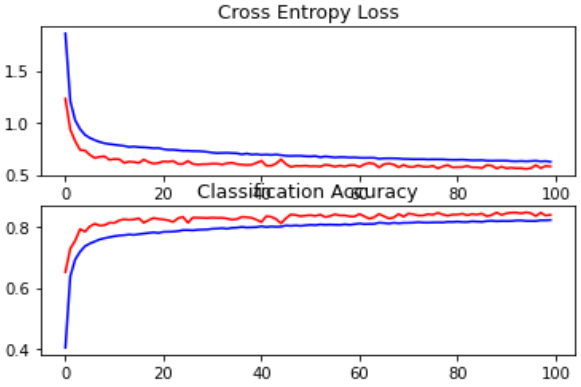


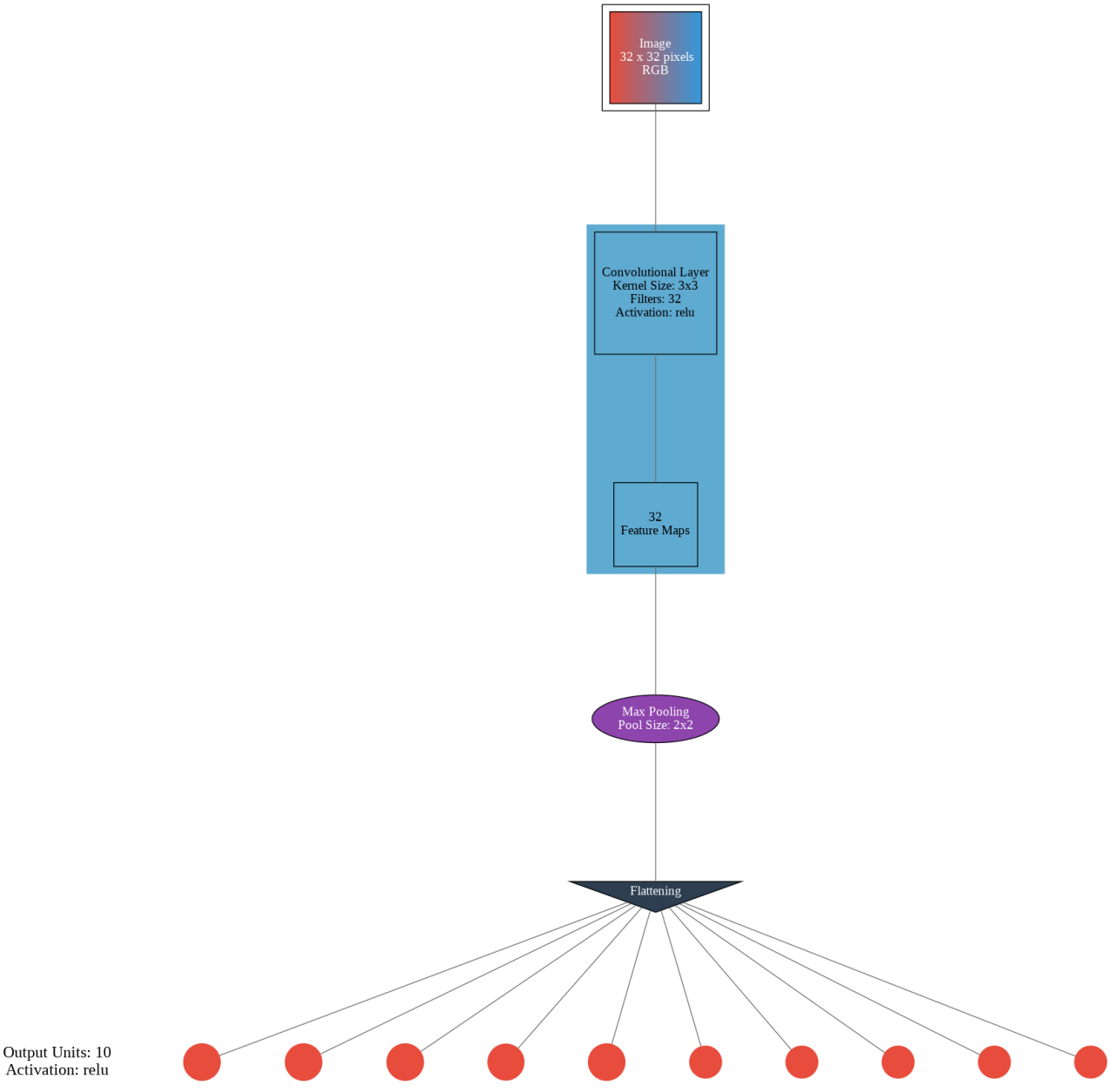
- In the figure above we have blue line as our training set and red line as our validation set and it shows that for training set model performed very well as from the accuracy graph, we can see in the last epochs model touched 96 whereas, red line which is for validation set just touched 89 mark on the graph which is because our model is not generalized and is highly overfitted.



2.4.3 Model 3 with single convoluted layer

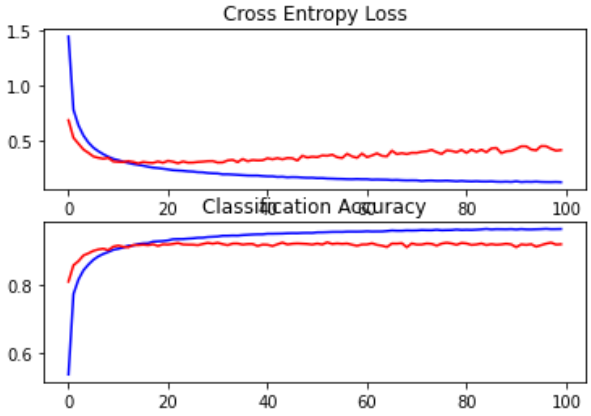
In Model 3 we kept only single convoluted layer and we found out that model is not learning much with just one layer and we saw underfitting. As the accuracy score for train set was just 82% and for test it was 82%. But we saw model did not learn much because of just single layer we model could not minimize the error function much, so it went to under fitting.

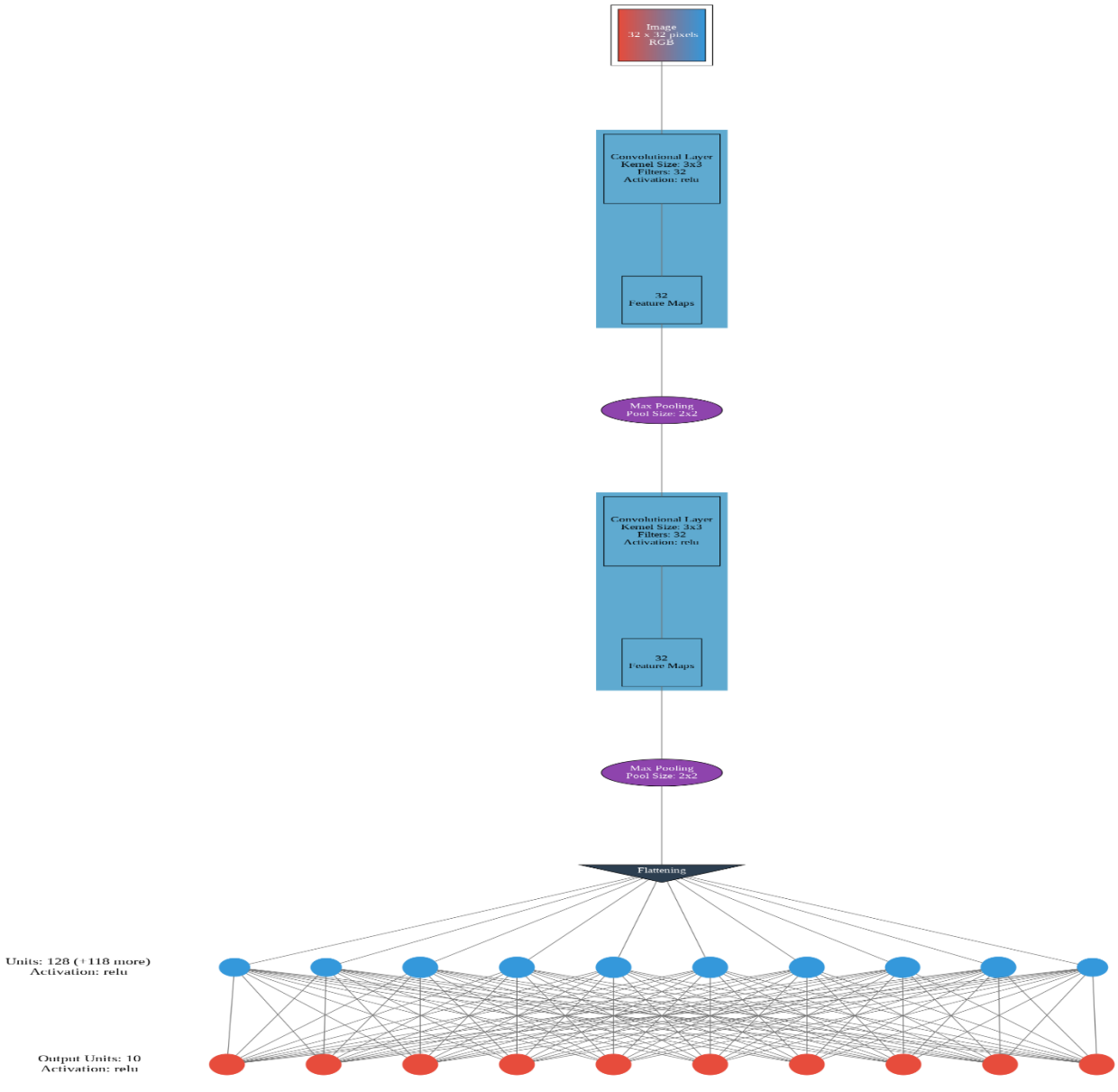




2.4.4 Model 4 with more convoluted layers

So, in this model we applied two convoluted layer each followed by a pooling layer and we got accuracy on training set for just below 97% which is very good but for testing set it was around 89% i.e. we can say model is overfitted but as we haven't generalize this neural network so it is expected that model will go into overfitting. So, we took this as our final model as it is already giving very good score in training set i.e. our model is failing to generalize and that will do in further steps using normalization.





2.4.5 Analysis on the results, such as overfitting/underfitting, variance bias

So, we saw with model 1 with sigmoid activations model was high biased and was not able to fit data well and in last model 4 with RELU activation and more convoluted layers model went to overfitting and having high variance.

For example, let us say our model instead of 10 classes have 2 classes or in short it is a binary classification just to understand better.

Figure 3

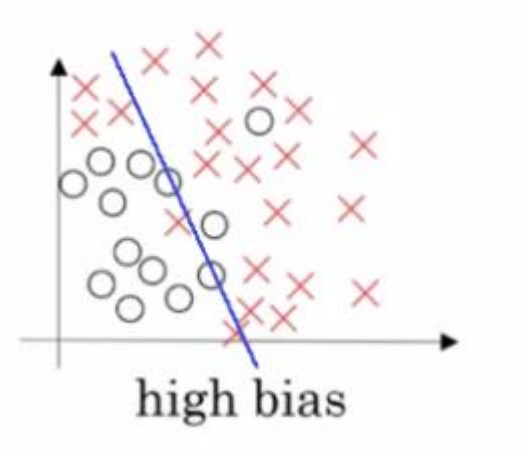


Figure 4

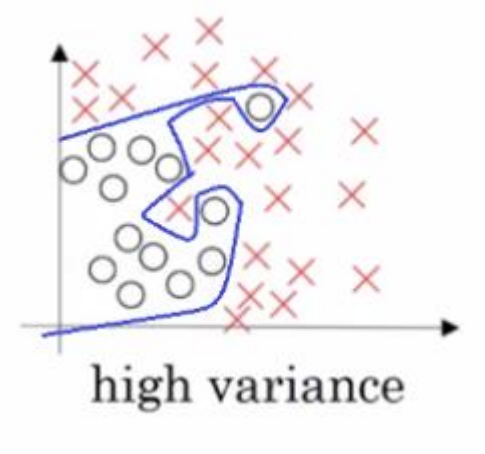
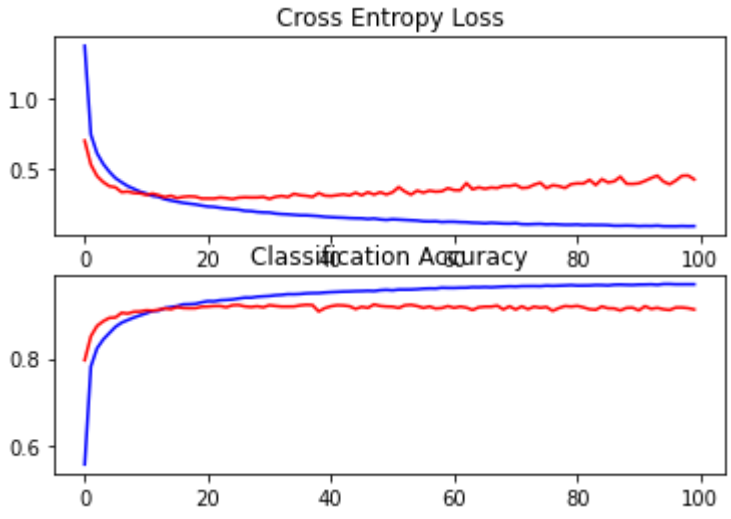


Image Credits: Rochak Agrawal [1]

Just for visual purposes we assume our problem is binary classifier than Model 1 showed something like figure 3 i.e. our model did not fitted well the data and was High biased and for Model 4 we can say showed something like figure 4 i.e. model showed overfitting of data and was high variance.

3.1 Describe the obtained results when changing the pooling operation.

Changing max pooling to average pooling improved our results but with not much significant values. Model with average pooling gave train and test accuracy of 96.97 and 89.97 which is just slight increase from Model with max pooling.




```
model.summary()
```

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 32, 32, 32)        896

 average_pooling2d (AverageP  (None, 16, 16, 32)       0
 ooling2D)

 conv2d_8 (Conv2D)           (None, 16, 16, 32)        9248

 average_pooling2d_1 (Averag  (None, 8, 8, 32)         0
 ePooling2D)

 flatten_4 (Flatten)         (None, 2048)              0

 dense_7 (Dense)             (None, 128)               262272

 dense_8 (Dense)             (None, 10)                1290

=================================================================
Total params: 273,706
Trainable params: 273,706
Non-trainable params: 0
_________________________________________________________________
```

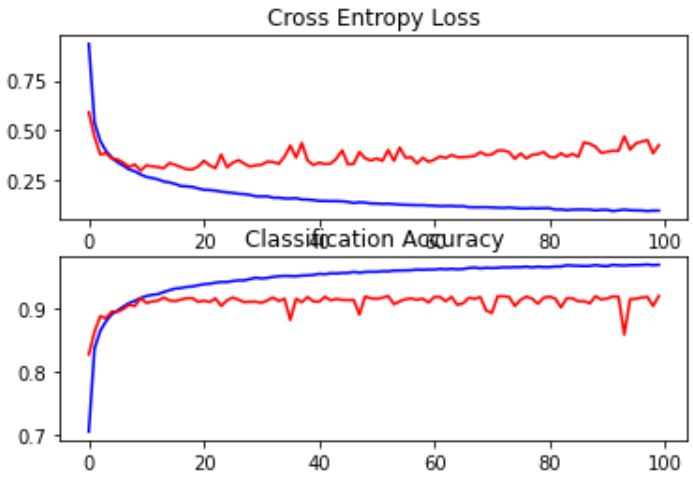
3.2 Do an analysis and describe conclusions and insights in terms of pooling operation. Which model provide the best performance?

Model	Train accuaracy	Test accuaracy
model 4 Final with more layers	96.32	89.866
Final Model with Average Pooling	96.97	89.978

From above results we can see that from max pooling to average pooling our model did not perform much better just a slight increase in accuracy. This can be because max pooling selects brighter pixels from the image and is useful when background of the image is dark. And as we are using cropped images it did not made much of a difference and hence, we get approximately same results.

3.3 Describe the obtained results when including the batch-normalization layer

After doing batch normalization we found train accuracy touching up to 97% and validation and test increased to 92% and 91% respectively.




```
model.summary()

Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 32, 32, 32)        896

 batch_normalization (BatchN  (None, 32, 32, 32)       128
 ormalization)

 max_pooling2d_5 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 16, 16, 32)        9248

 batch_normalization_1 (Batc  (None, 16, 16, 32)       128
 hNormalization)

 max_pooling2d_6 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 flatten_5 (Flatten)         (None, 2048)              0

 dense_9 (Dense)             (None, 128)               262272

 batch_normalization_2 (Batc  (None, 128)              512
 hNormalization)

 dense_10 (Dense)            (None, 10)                1290

=================================================================
Total params: 274,474
Trainable params: 274,090
Non-trainable params: 384
```

3.4 Do an analysis and describe conclusions and insights in terms of batch-normalization layer. Which model provide the best performance?

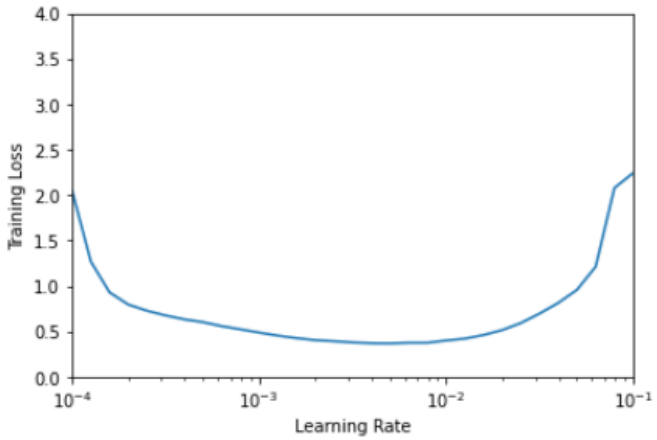
So, as we see we have seen an increase in accuracy as compared to our base Model 4, adding batch normalization and training weights with mean and variance instead of gradient descent helped our model to learn more. But still, we can see there is a huge gap between training and test set. Which shows like even after batch normalization our model is not generalised well and is facing issues while classifying test data or unseen data. So, to generalize more we performed L2 normalization as shown section 5 which helped generalizing our model very well.

4.1 Describe the obtained results after changing optimizer.

What we did is we tried finding the best learning rate by changing the learning rate in epoch by some value and then we plotted learning rate vs loss to find out at which learning rate we got the least loss.

```
#best model with optimizer finding best learning rate

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
#optimizer
lr_schedule = keras.callbacks.LearningRateScheduler(lambda epoch : 1e-4 * 10**(epoch/10))
optimizer = Adam(learning_rate= 1e-4, amsgrad=True)
# compile model
model.compile(optimizer=optimizer, loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val), callbacks = [lr_schedule])
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
[ ]  #plotting learning Rate vs Loss

     plot.semilogx(history.history['lr'],history.history['loss'])
     plot.axis([1e-4,1e-1,0,4])
     plot.xlabel('Learning Rate')
     plot.ylabel('Training Loss')
     plot.show()
```



4.2 Do an analysis and describe conclusions and insights in terms of the optimization

So, we can see from the graph that we got least loss around when learning rate is around 1e-3. We change learning rate in Adam optimizer to 1e-3 and results that we got were improved as compared to our baseline Model 4. We got training accuracy of around 96% and for validation and test we got 90% and 89% respectively. Which shows like our model improved as compared to our baseline model.

But still we can see that our model is high variance and is overfitted as we can see our model is performing good on training set but while categorizing test set our model struggles and shows much less accuracy as compared to train set. To overcome this and make our model less variance we will introduce Ridge regression or L2 norm regularization and dropout techniques which are used to generalize our models.

5.1 Describe the ideas or approaches you believe will improve your model and why?

5.1.1 L2 Regularization or Ridge regression

By analysing all the model, we have seen like we are getting accuracy for our training set very high as compared to our test set which means our model so far are not generalized and are overfitted with high variance.

So, to generalize and regularize our model we will introduce L2 regularization on our baseline Model 4. Why we are using L2 regularization or ridge regression because that is noting but feature selection and it helps us removing some features to take into consideration and our model will not be overfitted and will be generalized.

How can we say L2 regularization is feature selection?

$$minimize \frac{\lambda}{m} \sum_{i=1}^{n} ||W^T W x^{(i)} - x^{(i)}||^2 + \sum_{i=1}^{m} \sum_{i=1}^{k} g(W_j x^{(i)})$$

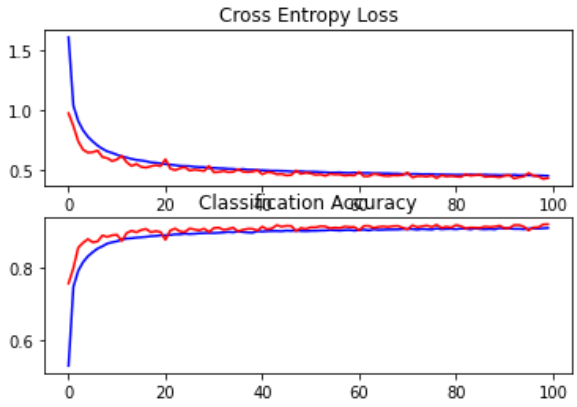
$W^T W x^{(i)}$ is nothing but $\hat{X}$

$x^{(i)}$ is nothing but X

$W_j x^i$ is nothing but output from neuron 'j' so we have to minimize this and to minimize o/p from neuron means we have to make this value zero. To make this value we have to minimize the sum as it is a summation, and we can make a sum zero by making terms in that summation as 0. And by saying that we are making terms in that summation zero we meant that we are not selecting those features hence L2 regularization is nothing but feature selection.

5.2 Describe the results on our base model.

After doing L2 regularization we got training set accuracy around 90% and validation and test set around 91% and 90% respectively. Which showed that our model did not overfitted on training data and for validation and test set we got same results as we got for our test set. Hence our model is regularized and more generalized which make our model less variance.



In the figure above we can see both are training and validation lines coincide and hence we can say our model is regularized and more generalized to unseen dataset.

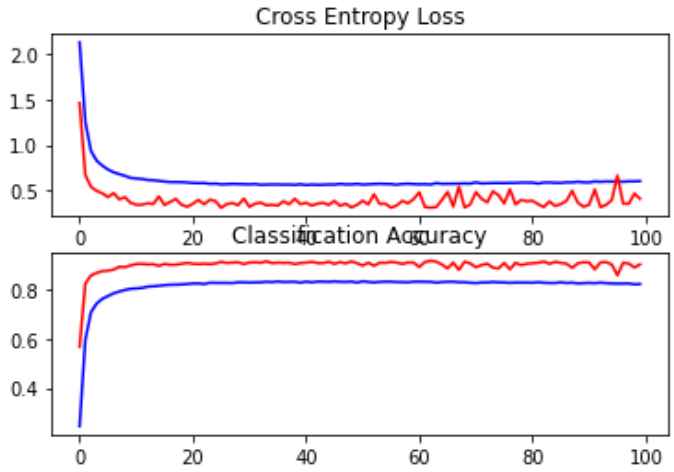
5.1.2 Dropout

In dropout while training our model we randomly dropout nodes during training. Some numbers of nodes are randomly ignored or dropped out. So, what happens in neural network is as the network is trained iteratively powerful connection which always fires gets more stronger and stronger as compared to weaker connection. And over many iterations only a fraction of nodes is trained, and rests stops participating.

So, ignoring nodes randomly makes our model more regularized and hence model is less variance and performs well on our test data.

5.2 Describe the results on our base model.

On our model, dropout showed very good results. Accuracy for training set came around 82.09% and for validation and test set it came around 90.03% and 89.578%. Which is more than our training set hence we can say that our model with dropouts is regularized and more generalized. That it why it showed good results with unseen dataset. Hence, we can say our model is less variance.



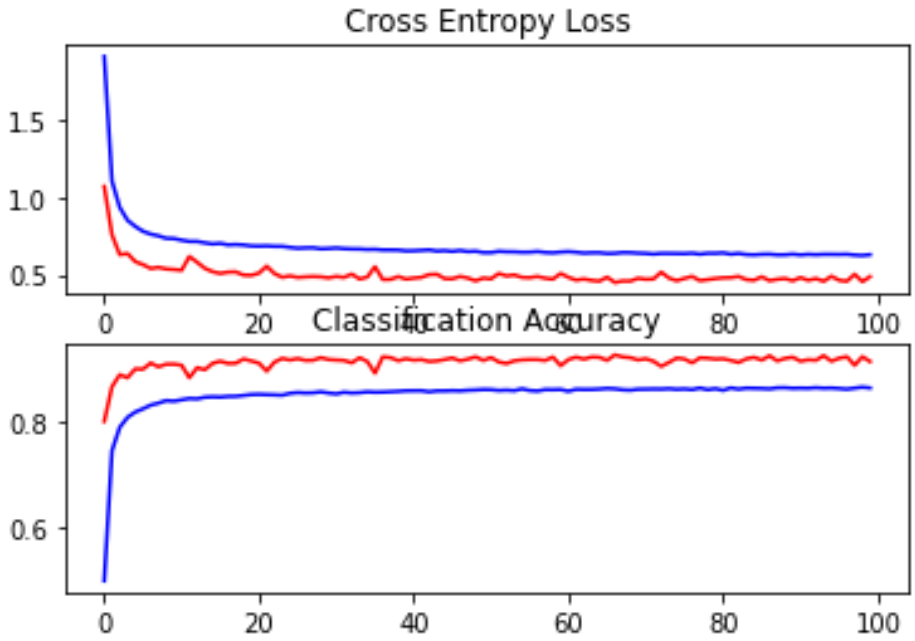
5.3 Describe conclusions and insights

In our final model, we combined all are findings from all above points.

- We used ReLU activation instead of sigmoid as sigmoid struggles with vanishing gradient problem.
- We introduced L2 regularization as we saw it helped countering the overfitting problem.
- We introduced dropouts, as we saw it helped in regularization and made our model more generalized.
- We introduced Batch Normalization also, though it showed very less results when we tune it on our base model but including it with other normalization techniques showed us that it can help regularizing our model further.
- We use Adam optimizer, which is very good gradient-based optimization of functions. With learning rate of 1e-3.

What results we got?

After integrating all things that we learnt, we found that our model did perform very well. And to be specific our model performed very well on unseen data. Let us analyse our results with below graph plot.



	Test Set	Validation set	Test set
Accuracy	86.31%	91.31%	91.26%

As we can see from the figure above, our validation set constantly performed well than our training set in each epoch. Which shows that our model is not overfitting. Which make our model less variance and more generalized for new and unseen dataset.

Summary and Conclusions

- After using Sigmoid as activation function, we found that our model went into underfitting. Because in backpropagation it showed vanishing gradient problem. And the model was highly biased.
- To solve the vanishing gradient problem, we introduced ReLU activation in our model 2 which helped solved the underfitting problem.
- We saw that in Model 3, with just single convoluted layer our model showed underfitting.
- In model 4, which we took as our baseline model. We found that our model is performing best on training set but struggling on test set. Hence our model is having high variance.
- We tried Average pooling instead of max pooling on our baseline model, but it did not make much of a difference for our dataset.
- We found that with batch-normalization our model did perform a little better than our baseline model but still we saw overfitting in our model.
- We found very good results after L2 normalization, as it helped us countering the overfitting problem and made our model more generalized. And we also saw how L2 regularization is nothing but feature selection.
- We saw how dropout can be used for regularizing our model and making our model less variance.
- We found that 1e-3 is the best learning rate for our model.

Models Accuracy Summary:

Model	**Train accuaracy**	**Test accuaracy**
model 1 with softmax activations	89.92	87.085
model 2 with relu activations	95.11	88.975
model 3 with only 1 Conv2D layer	82.3	82.522
model 4 Final with more layers	96.32	89.866
Final Model with Average Pooling	96.97	89.978
Final Model with Batch Normalization	97.05	90.915
Final Model with L2 normalization	**90.72**	**90.619**
Final Model with Dropout	**82.09**	**89.578**
Final model with best learning rate 1e-3	96.15	89.006
Final Model with all hyper tunings	**86.31**	**91.261**

References

[1] Rochak Agrawal. Improving Deep Neural Networks. https://towardsdatascience.com/improving-deep-neural-networks-b5984e29e336, 2019. Published: 2019-06-19.

[2] Anthony J. Bell and Terrence J. Sejnowski. The “independent components” of natural scenes are edgefilters.Vision Research, 37(23):3327–3338, 1997.

[3] Adam Coates, A. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised featurelearning. InAISTATS, 2011.

[4] Yuval Netzer, Tiejie Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[5] Dimitrios roussis. SVHN Classification with CNN. https://www.kaggle.com/dimitriosroussis/svhn-classification-with-cnn-keras-96-acc/notebook, 2019. Published: 2019-06-19.

In [1]:
```
pip install keras_visualizer
```

```
Collecting keras_visualizer
  Downloading keras_visualizer-2.4-py3-none-any.whl (5.4 kB)
Installing collected packages: keras-visualizer
Successfully installed keras-visualizer-2.4
```

In [22]:
```
import numpy as np
import keras
import cv2
from IPython.display import Image
from scipy.io import loadmat
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
from matplotlib import pyplot as plot
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import AveragePooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import BatchNormalization
from keras.optimizer_v1 import sgd
from keras_visualizer import visualizer
from keras.regularizers import l2
from keras.optimizer_v1 import adam
from tensorflow.keras.optimizers import Adam,SGD
```

In [4]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In [5]:
```
train_rawdata = loadmat('/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/train_32x32.mat')
test_rawdata = loadmat('/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/test_32x32.mat')
```

In [6]:
```
train_imgs = np.array(train_rawdata['X'])
test_imgs = np.array(test_rawdata['X'])

train_labels = train_rawdata['y']
test_labels = test_rawdata['y']
```

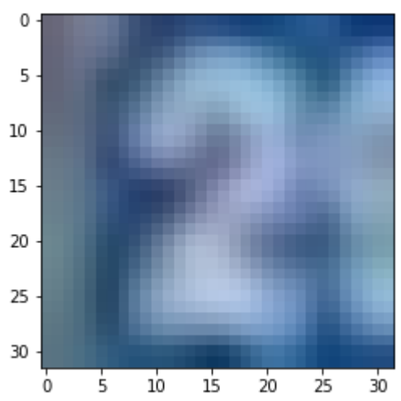
In [7]:
```
print(train_imgs.shape)
print(test_imgs.shape)
```

```
(32, 32, 3, 73257)
(32, 32, 3, 26032)
```

In [8]:
```
train_imgs = np.moveaxis(train_imgs, -1, 0)
test_imgs = np.moveaxis(test_imgs, -1, 0)

print(train_imgs.shape)
print(test_imgs.shape)
```

```
(73257, 32, 32, 3)
(26032, 32, 32, 3)
```

In [9]:
```
plot.imshow(train_imgs[10522])
plot.show()
print('Label: ', train_labels[10522])
```

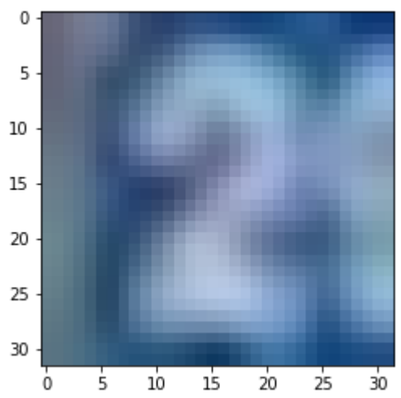



```
Label:  [2]
```

In [10]:
```
#Normalization

train_imgs = train_imgs.astype('float64')
test_imgs = test_imgs.astype('float64')

train_labels = train_labels.astype('int64')
test_labels = test_labels.astype('int64')

train_imgs /= 255.0
test_imgs /= 255.0
```

In [11]:
```
# to create label of images like 0 0 0 1 0 0 0 0 0 0 = 4
lb = LabelBinarizer()
train_labels = lb.fit_transform(train_labels)
test_labels = lb.fit_transform(test_labels)
```

In [12]:
```
plot.imshow(train_imgs[10522])
plot.show()
print('Label: ', train_labels[10522])
```




```
Label:  [0 1 0 0 0 0 0 0 0 0]
```

In [13]:
```
#data expansion
train_datagen = ImageDataGenerator(rotation_range=8,
```

```
                                zoom_range=[0.95, 1.05],
                                height_shift_range=0.10,
                                shear_range=0.15)
```

In [14]:
```
X_train, X_val, y_train, y_val = train_test_split(train_imgs, train_labels, test_size=0.15, random_state=22)
```

In [15]:
```
def summarize_outcome(history):
    # plot loss
    plot.subplot(211)
    plot.title('Cross Entropy Loss')
    plot.plot(history.history['loss'], color='blue', label='train')
    plot.plot(history.history['val_loss'], color='red', label='validation')
    # plot accuracy
    plot.subplot(212)
    plot.title('Classification Accuracy')
    plot.plot(history.history['accuracy'], color='blue', label='train')
    plot.plot(history.history['val_accuracy'], color='red', label='validation')
    plot.show()
    plot.close()
```

CNN Models

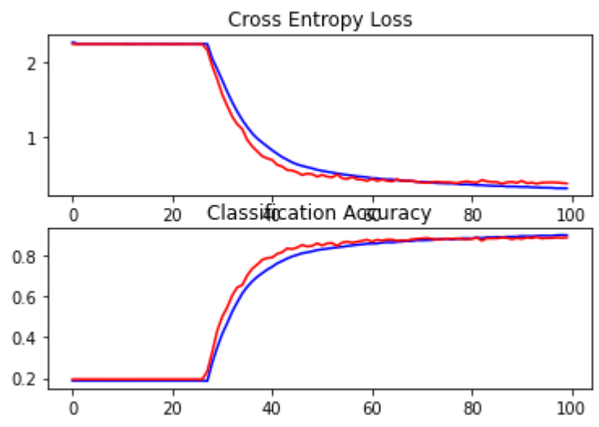
In [ ]:
```
#model 1 with sigmoid activations

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='sigmoid', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(Conv2D(32, (3, 3), activation='sigmoid', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(68, activation='sigmoid', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 37s 74ms/step - loss: 2.2569 - accuracy: 0.1879 - val_loss: 2.2345 - val_accuracy: 0.1949
Epoch 2/100
487/487 [==============================] - 35s 72ms/step - loss: 2.2391 - accuracy: 0.1882 - val_loss: 2.2329 - val_accuracy: 0.1949
Epoch 3/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2361 - val_accuracy: 0.1949
Epoch 4/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2339 - val_accuracy: 0.1949
Epoch 5/100
487/487 [==============================] - 34s 70ms/step - loss: 2.2390 - accuracy: 0.1882 - val_loss: 2.2361 - val_accuracy: 0.1949
Epoch 6/100
487/487 [==============================] - 34s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2335 - val_accuracy: 0.1949
Epoch 7/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2353 - val_accuracy: 0.1949
Epoch 8/100
487/487 [==============================] - 34s 71ms/step - loss: 2.2393 - accuracy: 0.1882 - val_loss: 2.2342 - val_accuracy: 0.1949
Epoch 9/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2392 - accuracy: 0.1880 - val_loss: 2.2342 - val_accuracy: 0.1949
Epoch 10/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2341 - val_accuracy: 0.1949
Epoch 11/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2390 - accuracy: 0.1882 - val_loss: 2.2348 - val_accuracy: 0.1949
Epoch 12/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2393 - accuracy: 0.1881 - val_loss: 2.2326 - val_accuracy: 0.1949
Epoch 13/100
487/487 [==============================] - 34s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2330 - val_accuracy: 0.1949
Epoch 14/100
487/487 [==============================] - 34s 70ms/step - loss: 2.2390 - accuracy: 0.1882 - val_loss: 2.2348 - val_accuracy: 0.1949
Epoch 15/100
487/487 [==============================] - 34s 70ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2336 - val_accuracy: 0.1949
Epoch 16/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2391 - accuracy: 0.1882 - val_loss: 2.2342 - val_accuracy: 0.1949
Epoch 17/100
487/487 [==============================] - 34s 71ms/step - loss: 2.2394 - accuracy: 0.1882 - val_loss: 2.2333 - val_accuracy: 0.1949
Epoch 18/100
487/487 [==============================] - 35s 72ms/step - loss: 2.2393 - accuracy: 0.1882 - val_loss: 2.2354 - val_accuracy: 0.1949
Epoch 19/100
487/487 [==============================] - 35s 72ms/step - loss: 2.2391 - accuracy: 0.1882 - val_loss: 2.2339 - val_accuracy: 0.1949
Epoch 20/100
487/487 [==============================] - 35s 72ms/step - loss: 2.2391 - accuracy: 0.1882 - val_loss: 2.2329 - val_accuracy: 0.1949
Epoch 21/100
487/487 [==============================] - 35s 72ms/step - loss: 2.2391 - accuracy: 0.1882 - val_loss: 2.2328 - val_accuracy: 0.1949
Epoch 22/100
487/487 [==============================] - 34s 70ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2333 - val_accuracy: 0.1949
Epoch 23/100
487/487 [==============================] - 34s 70ms/step - loss: 2.2393 - accuracy: 0.1882 - val_loss: 2.2338 - val_accuracy: 0.1949
Epoch 24/100
487/487 [==============================] - 34s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2338 - val_accuracy: 0.1949
Epoch 25/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2392 - accuracy: 0.1882 - val_loss: 2.2353 - val_accuracy: 0.1949
Epoch 26/100
487/487 [==============================] - 34s 70ms/step - loss: 2.2390 - accuracy: 0.1882 - val_loss: 2.2336 - val_accuracy: 0.1949
Epoch 27/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2394 - accuracy: 0.1879 - val_loss: 2.2328 - val_accuracy: 0.1949
Epoch 28/100
487/487 [==============================] - 35s 71ms/step - loss: 2.2344 - accuracy: 0.1867 - val_loss: 2.1556 - val_accuracy: 0.2342
Epoch 29/100
487/487 [==============================] - 35s 71ms/step - loss: 2.0388 - accuracy: 0.2777 - val_loss: 1.9417 - val_accuracy: 0.3301
Epoch 30/100
487/487 [==============================] - 35s 72ms/step - loss: 1.9001 - accuracy: 0.3520 - val_loss: 1.7657 - val_accuracy: 0.4346
Epoch 31/100
487/487 [==============================] - 35s 72ms/step - loss: 1.7530 - accuracy: 0.4180 - val_loss: 1.5655 - val_accuracy: 0.5019
Epoch 32/100
487/487 [==============================] - 35s 72ms/step - loss: 1.6026 - accuracy: 0.4705 - val_loss: 1.4153 - val_accuracy: 0.5476
Epoch 33/100
487/487 [==============================] - 35s 71ms/step - loss: 1.4642 - accuracy: 0.5266 - val_loss: 1.2736 - val_accuracy: 0.6060
Epoch 34/100
487/487 [==============================] - 34s 70ms/step - loss: 1.3411 - accuracy: 0.5719 - val_loss: 1.1750 - val_accuracy: 0.6442
Epoch 35/100
487/487 [==============================] - 34s 70ms/step - loss: 1.2317 - accuracy: 0.6149 - val_loss: 1.1107 - val_accuracy: 0.6558
Epoch 36/100
487/487 [==============================] - 34s 69ms/step - loss: 1.1381 - accuracy: 0.6473 - val_loss: 0.9674 - val_accuracy: 0.7035
Epoch 37/100
487/487 [==============================] - 34s 70ms/step - loss: 1.0538 - accuracy: 0.6742 - val_loss: 0.8821 - val_accuracy: 0.7358
Epoch 38/100
487/487 [==============================] - 33s 69ms/step - loss: 0.9866 - accuracy: 0.6959 - val_loss: 0.8165 - val_accuracy: 0.7561
Epoch 39/100
487/487 [==============================] - 35s 71ms/step - loss: 0.9335 - accuracy: 0.7140 - val_loss: 0.7529 - val_accuracy: 0.7793
Epoch 40/100
487/487 [==============================] - 35s 72ms/step - loss: 0.8817 - accuracy: 0.7311 - val_loss: 0.7240 - val_accuracy: 0.7879
Epoch 41/100
487/487 [==============================] - 35s 71ms/step - loss: 0.8312 - accuracy: 0.7454 - val_loss: 0.7039 - val_accuracy: 0.7902
Epoch 42/100
487/487 [==============================] - 35s 71ms/step - loss: 0.7852 - accuracy: 0.7619 - val_loss: 0.6378 - val_accuracy: 0.8086
Epoch 43/100
487/487 [==============================] - 35s 72ms/step - loss: 0.7413 - accuracy: 0.7734 - val_loss: 0.6174 - val_accuracy: 0.8166
Epoch 44/100
487/487 [==============================] - 35s 72ms/step - loss: 0.7064 - accuracy: 0.7853 - val_loss: 0.5697 - val_accuracy: 0.8347
```

```
Epoch 45/100
487/487 [==============================] - 35s 72ms/step - loss: 0.6712 - accuracy: 0.7956 - val_loss: 0.5650 - val_accuracy: 0.8312
Epoch 46/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6444 - accuracy: 0.8031 - val_loss: 0.5401 - val_accuracy: 0.8394
Epoch 47/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6253 - accuracy: 0.8114 - val_loss: 0.5038 - val_accuracy: 0.8514
Epoch 48/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6083 - accuracy: 0.8154 - val_loss: 0.5232 - val_accuracy: 0.8438
Epoch 49/100
487/487 [==============================] - 34s 70ms/step - loss: 0.5927 - accuracy: 0.8205 - val_loss: 0.5165 - val_accuracy: 0.8474
Epoch 50/100
487/487 [==============================] - 34s 70ms/step - loss: 0.5730 - accuracy: 0.8267 - val_loss: 0.4820 - val_accuracy: 0.8595
Epoch 51/100
487/487 [==============================] - 34s 70ms/step - loss: 0.5601 - accuracy: 0.8302 - val_loss: 0.5066 - val_accuracy: 0.8490
Epoch 52/100
487/487 [==============================] - 34s 70ms/step - loss: 0.5481 - accuracy: 0.8334 - val_loss: 0.4830 - val_accuracy: 0.8591
Epoch 53/100
487/487 [==============================] - 34s 69ms/step - loss: 0.5368 - accuracy: 0.8370 - val_loss: 0.4744 - val_accuracy: 0.8609
Epoch 54/100
487/487 [==============================] - 34s 70ms/step - loss: 0.5254 - accuracy: 0.8402 - val_loss: 0.5122 - val_accuracy: 0.8473
Epoch 55/100
487/487 [==============================] - 35s 72ms/step - loss: 0.5150 - accuracy: 0.8425 - val_loss: 0.4614 - val_accuracy: 0.8635
Epoch 56/100
487/487 [==============================] - 34s 70ms/step - loss: 0.5053 - accuracy: 0.8462 - val_loss: 0.4524 - val_accuracy: 0.8679
Epoch 57/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4975 - accuracy: 0.8490 - val_loss: 0.4781 - val_accuracy: 0.8627
Epoch 58/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4873 - accuracy: 0.8525 - val_loss: 0.4482 - val_accuracy: 0.8673
Epoch 59/100
487/487 [==============================] - 35s 71ms/step - loss: 0.4819 - accuracy: 0.8545 - val_loss: 0.4424 - val_accuracy: 0.8717
Epoch 60/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4745 - accuracy: 0.8574 - val_loss: 0.4265 - val_accuracy: 0.8771
Epoch 61/100
487/487 [==============================] - 35s 71ms/step - loss: 0.4625 - accuracy: 0.8584 - val_loss: 0.4543 - val_accuracy: 0.8675
Epoch 62/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4611 - accuracy: 0.8588 - val_loss: 0.4268 - val_accuracy: 0.8770
Epoch 63/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4491 - accuracy: 0.8633 - val_loss: 0.4445 - val_accuracy: 0.8727
Epoch 64/100
487/487 [==============================] - 35s 71ms/step - loss: 0.4467 - accuracy: 0.8647 - val_loss: 0.4283 - val_accuracy: 0.8780
Epoch 65/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4460 - accuracy: 0.8638 - val_loss: 0.4200 - val_accuracy: 0.8774
Epoch 66/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4376 - accuracy: 0.8667 - val_loss: 0.4499 - val_accuracy: 0.8722
Epoch 67/100
487/487 [==============================] - 34s 71ms/step - loss: 0.4320 - accuracy: 0.8685 - val_loss: 0.4214 - val_accuracy: 0.8744
Epoch 68/100
487/487 [==============================] - 34s 71ms/step - loss: 0.4253 - accuracy: 0.8709 - val_loss: 0.4313 - val_accuracy: 0.8729
Epoch 69/100
487/487 [==============================] - 34s 69ms/step - loss: 0.4251 - accuracy: 0.8710 - val_loss: 0.4238 - val_accuracy: 0.8796
Epoch 70/100
487/487 [==============================] - 34s 69ms/step - loss: 0.4135 - accuracy: 0.8746 - val_loss: 0.4166 - val_accuracy: 0.8811
Epoch 71/100
487/487 [==============================] - 34s 70ms/step - loss: 0.4156 - accuracy: 0.8741 - val_loss: 0.4036 - val_accuracy: 0.8837
Epoch 72/100
487/487 [==============================] - 34s 69ms/step - loss: 0.4079 - accuracy: 0.8752 - val_loss: 0.4016 - val_accuracy: 0.8855
Epoch 73/100
487/487 [==============================] - 34s 69ms/step - loss: 0.4046 - accuracy: 0.8755 - val_loss: 0.4089 - val_accuracy: 0.8817
Epoch 74/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3977 - accuracy: 0.8791 - val_loss: 0.4175 - val_accuracy: 0.8773
Epoch 75/100
487/487 [==============================] - 34s 69ms/step - loss: 0.3956 - accuracy: 0.8808 - val_loss: 0.4077 - val_accuracy: 0.8806
Epoch 76/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3926 - accuracy: 0.8818 - val_loss: 0.4088 - val_accuracy: 0.8828
Epoch 77/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3930 - accuracy: 0.8791 - val_loss: 0.4050 - val_accuracy: 0.8813
Epoch 78/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3880 - accuracy: 0.8815 - val_loss: 0.3970 - val_accuracy: 0.8841
Epoch 79/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3796 - accuracy: 0.8844 - val_loss: 0.4123 - val_accuracy: 0.8800
Epoch 80/100
487/487 [==============================] - 35s 72ms/step - loss: 0.3813 - accuracy: 0.8844 - val_loss: 0.4212 - val_accuracy: 0.8791
Epoch 81/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3768 - accuracy: 0.8835 - val_loss: 0.4086 - val_accuracy: 0.8821
Epoch 82/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3709 - accuracy: 0.8869 - val_loss: 0.4030 - val_accuracy: 0.8874
Epoch 83/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3714 - accuracy: 0.8853 - val_loss: 0.4395 - val_accuracy: 0.8723
Epoch 84/100
487/487 [==============================] - 34s 71ms/step - loss: 0.3650 - accuracy: 0.8890 - val_loss: 0.4192 - val_accuracy: 0.8842
Epoch 85/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3623 - accuracy: 0.8889 - val_loss: 0.4130 - val_accuracy: 0.8834
Epoch 86/100
487/487 [==============================] - 34s 71ms/step - loss: 0.3601 - accuracy: 0.8907 - val_loss: 0.4123 - val_accuracy: 0.8837
Epoch 87/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3604 - accuracy: 0.8897 - val_loss: 0.3882 - val_accuracy: 0.8883
Epoch 88/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3527 - accuracy: 0.8920 - val_loss: 0.4083 - val_accuracy: 0.8860
Epoch 89/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3525 - accuracy: 0.8917 - val_loss: 0.4138 - val_accuracy: 0.8797
Epoch 90/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3510 - accuracy: 0.8924 - val_loss: 0.4011 - val_accuracy: 0.8850
Epoch 91/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3475 - accuracy: 0.8951 - val_loss: 0.4276 - val_accuracy: 0.8812
Epoch 92/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3479 - accuracy: 0.8945 - val_loss: 0.3956 - val_accuracy: 0.8879
Epoch 93/100
487/487 [==============================] - 35s 71ms/step - loss: 0.3429 - accuracy: 0.8942 - val_loss: 0.4041 - val_accuracy: 0.8847
Epoch 94/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3405 - accuracy: 0.8958 - val_loss: 0.3897 - val_accuracy: 0.8903
Epoch 95/100
487/487 [==============================] - 34s 71ms/step - loss: 0.3371 - accuracy: 0.8972 - val_loss: 0.4035 - val_accuracy: 0.8831
Epoch 96/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3367 - accuracy: 0.8965 - val_loss: 0.4055 - val_accuracy: 0.8857
Epoch 97/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3349 - accuracy: 0.8963 - val_loss: 0.4075 - val_accuracy: 0.8851
Epoch 98/100
487/487 [==============================] - 33s 69ms/step - loss: 0.3292 - accuracy: 0.8994 - val_loss: 0.4044 - val_accuracy: 0.8881
Epoch 99/100
487/487 [==============================] - 34s 69ms/step - loss: 0.3298 - accuracy: 0.9003 - val_loss: 0.3994 - val_accuracy: 0.8864
Epoch 100/100
487/487 [==============================] - 34s 70ms/step - loss: 0.3288 - accuracy: 0.8992 - val_loss: 0.3932 - val_accuracy: 0.8877
> accuaracy of test set 87.085
```

In [ ]:

```
summarize_outcome(history = history)
```

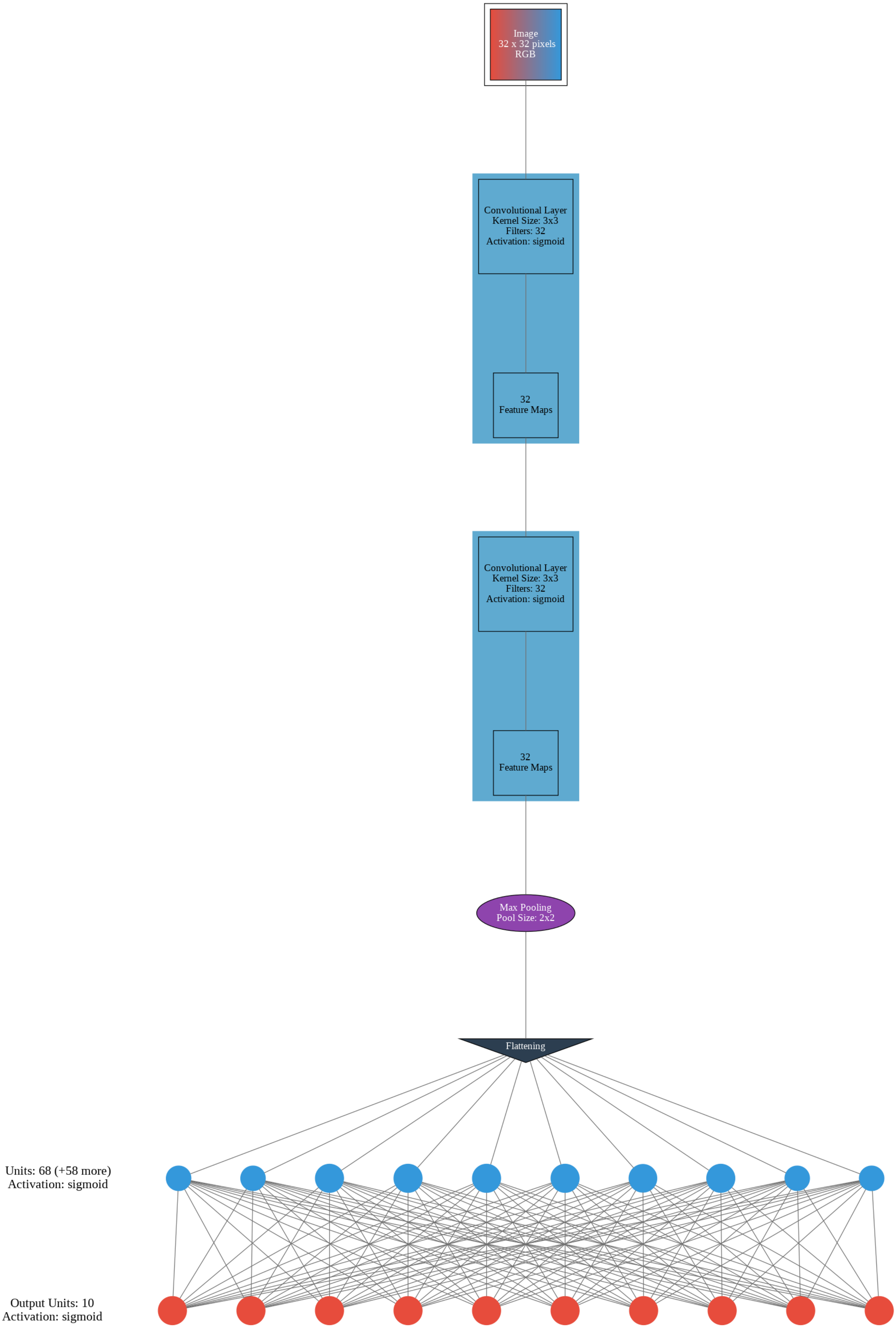


In [ ]:
```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 32, 32, 32)        896

 conv2d_7 (Conv2D)           (None, 32, 32, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 flatten_3 (Flatten)         (None, 8192)              0

 dense_4 (Dense)             (None, 68)                557124

 dense_5 (Dense)             (None, 10)                690

=================================================================
Total params: 567,958
Trainable params: 567,958
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:
```
visualizer(model,format='png',view=True, filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model1')
Image(filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model1.png')
```

Out[ ]:



In [ ]:

```
#model 2 with relu activations

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
```

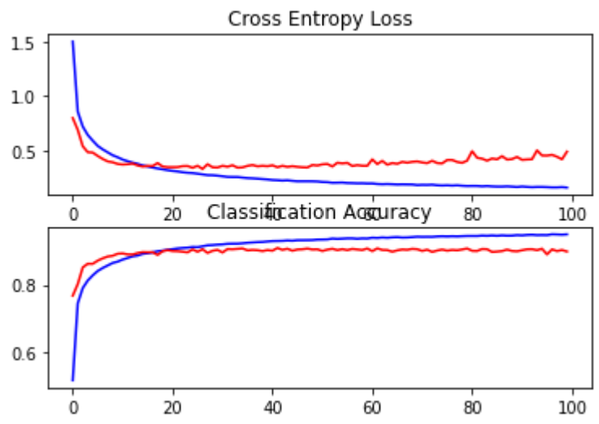
```
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(68, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 54ms/step - loss: 1.4975 - accuracy: 0.5145 - val_loss: 0.7980 - val_accuracy: 0.7677
Epoch 2/100
487/487 [==============================] - 27s 54ms/step - loss: 0.8573 - accuracy: 0.7446 - val_loss: 0.6903 - val_accuracy: 0.8022
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7207 - accuracy: 0.7903 - val_loss: 0.5401 - val_accuracy: 0.8520
Epoch 4/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6433 - accuracy: 0.8129 - val_loss: 0.4854 - val_accuracy: 0.8632
Epoch 5/100
487/487 [==============================] - 27s 55ms/step - loss: 0.5912 - accuracy: 0.8284 - val_loss: 0.4820 - val_accuracy: 0.8629
Epoch 6/100
487/487 [==============================] - 27s 54ms/step - loss: 0.5431 - accuracy: 0.8415 - val_loss: 0.4510 - val_accuracy: 0.8725
Epoch 7/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5111 - accuracy: 0.8505 - val_loss: 0.4233 - val_accuracy: 0.8793
Epoch 8/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4839 - accuracy: 0.8580 - val_loss: 0.4009 - val_accuracy: 0.8848
Epoch 9/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4572 - accuracy: 0.8655 - val_loss: 0.3931 - val_accuracy: 0.8871
Epoch 10/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4389 - accuracy: 0.8695 - val_loss: 0.3776 - val_accuracy: 0.8932
Epoch 11/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4178 - accuracy: 0.8757 - val_loss: 0.3725 - val_accuracy: 0.8941
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4027 - accuracy: 0.8804 - val_loss: 0.3753 - val_accuracy: 0.8911
Epoch 13/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3895 - accuracy: 0.8854 - val_loss: 0.3811 - val_accuracy: 0.8908
Epoch 14/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3783 - accuracy: 0.8877 - val_loss: 0.3639 - val_accuracy: 0.8952
Epoch 15/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3640 - accuracy: 0.8927 - val_loss: 0.3539 - val_accuracy: 0.8979
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3552 - accuracy: 0.8947 - val_loss: 0.3599 - val_accuracy: 0.8974
Epoch 17/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3450 - accuracy: 0.8973 - val_loss: 0.3587 - val_accuracy: 0.8977
Epoch 18/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3362 - accuracy: 0.9006 - val_loss: 0.3858 - val_accuracy: 0.8895
Epoch 19/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3285 - accuracy: 0.9024 - val_loss: 0.3540 - val_accuracy: 0.9008
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3207 - accuracy: 0.9050 - val_loss: 0.3482 - val_accuracy: 0.9039
Epoch 21/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3141 - accuracy: 0.9069 - val_loss: 0.3481 - val_accuracy: 0.9013
Epoch 22/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3090 - accuracy: 0.9090 - val_loss: 0.3486 - val_accuracy: 0.9008
Epoch 23/100
487/487 [==============================] - 27s 54ms/step - loss: 0.3017 - accuracy: 0.9105 - val_loss: 0.3575 - val_accuracy: 0.8993
Epoch 24/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2970 - accuracy: 0.9113 - val_loss: 0.3587 - val_accuracy: 0.8972
Epoch 25/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2943 - accuracy: 0.9133 - val_loss: 0.3423 - val_accuracy: 0.9060
Epoch 26/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2884 - accuracy: 0.9128 - val_loss: 0.3612 - val_accuracy: 0.8991
Epoch 27/100
487/487 [==============================] - 27s 56ms/step - loss: 0.2819 - accuracy: 0.9155 - val_loss: 0.3317 - val_accuracy: 0.9067
Epoch 28/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2755 - accuracy: 0.9183 - val_loss: 0.3753 - val_accuracy: 0.8954
Epoch 29/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2746 - accuracy: 0.9190 - val_loss: 0.3472 - val_accuracy: 0.9020
Epoch 30/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2705 - accuracy: 0.9203 - val_loss: 0.3451 - val_accuracy: 0.9046
Epoch 31/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2649 - accuracy: 0.9211 - val_loss: 0.3606 - val_accuracy: 0.8968
Epoch 32/100
487/487 [==============================] - 27s 54ms/step - loss: 0.2589 - accuracy: 0.9229 - val_loss: 0.3523 - val_accuracy: 0.9072
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2580 - accuracy: 0.9233 - val_loss: 0.3656 - val_accuracy: 0.9065
Epoch 34/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2578 - accuracy: 0.9233 - val_loss: 0.3440 - val_accuracy: 0.9078
Epoch 35/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2529 - accuracy: 0.9245 - val_loss: 0.3462 - val_accuracy: 0.9090
Epoch 36/100
487/487 [==============================] - 27s 54ms/step - loss: 0.2484 - accuracy: 0.9260 - val_loss: 0.3601 - val_accuracy: 0.9033
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2461 - accuracy: 0.9269 - val_loss: 0.3680 - val_accuracy: 0.9044
Epoch 38/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2429 - accuracy: 0.9279 - val_loss: 0.3561 - val_accuracy: 0.9036
Epoch 39/100
487/487 [==============================] - 27s 54ms/step - loss: 0.2392 - accuracy: 0.9284 - val_loss: 0.3605 - val_accuracy: 0.9012
Epoch 40/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2357 - accuracy: 0.9298 - val_loss: 0.3570 - val_accuracy: 0.9040
Epoch 41/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2307 - accuracy: 0.9310 - val_loss: 0.3651 - val_accuracy: 0.9023
Epoch 42/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2286 - accuracy: 0.9313 - val_loss: 0.3476 - val_accuracy: 0.9104
Epoch 43/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2255 - accuracy: 0.9317 - val_loss: 0.3604 - val_accuracy: 0.9048
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2278 - accuracy: 0.9328 - val_loss: 0.3510 - val_accuracy: 0.9081
Epoch 45/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2229 - accuracy: 0.9321 - val_loss: 0.3571 - val_accuracy: 0.9022
Epoch 46/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2189 - accuracy: 0.9336 - val_loss: 0.3510 - val_accuracy: 0.9058
Epoch 47/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2192 - accuracy: 0.9337 - val_loss: 0.3471 - val_accuracy: 0.9065
Epoch 48/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2191 - accuracy: 0.9338 - val_loss: 0.3454 - val_accuracy: 0.9086
Epoch 49/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2187 - accuracy: 0.9341 - val_loss: 0.3681 - val_accuracy: 0.9058
Epoch 50/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2171 - accuracy: 0.9354 - val_loss: 0.3636 - val_accuracy: 0.9064
Epoch 51/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2124 - accuracy: 0.9356 - val_loss: 0.3741 - val_accuracy: 0.9014
Epoch 52/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2108 - accuracy: 0.9361 - val_loss: 0.3773 - val_accuracy: 0.9054
Epoch 53/100
487/487 [==============================] - 27s 54ms/step - loss: 0.2054 - accuracy: 0.9385 - val_loss: 0.3545 - val_accuracy: 0.9080
Epoch 54/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2062 - accuracy: 0.9376 - val_loss: 0.3885 - val_accuracy: 0.9040
Epoch 55/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2074 - accuracy: 0.9383 - val_loss: 0.3827 - val_accuracy: 0.9068
Epoch 56/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2034 - accuracy: 0.9392 - val_loss: 0.3875 - val_accuracy: 0.9040
Epoch 57/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2024 - accuracy: 0.9392 - val_loss: 0.3587 - val_accuracy: 0.9075
Epoch 58/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2005 - accuracy: 0.9380 - val_loss: 0.3671 - val_accuracy: 0.9058
Epoch 59/100
487/487 [==============================] - 27s 54ms/step - loss: 0.2010 - accuracy: 0.9391 - val_loss: 0.3612 - val_accuracy: 0.9043
Epoch 60/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1988 - accuracy: 0.9387 - val_loss: 0.3605 - val_accuracy: 0.9089
Epoch 61/100
```

```
487/487 [==============================] - 26s 54ms/step - loss: 0.1992 - accuracy: 0.9412 - val_loss: 0.4198 - val_accuracy: 0.9009
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1941 - accuracy: 0.9404 - val_loss: 0.3764 - val_accuracy: 0.9092
Epoch 63/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1920 - accuracy: 0.9418 - val_loss: 0.4058 - val_accuracy: 0.9044
Epoch 64/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1937 - accuracy: 0.9409 - val_loss: 0.3706 - val_accuracy: 0.9041
Epoch 65/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1919 - accuracy: 0.9422 - val_loss: 0.3858 - val_accuracy: 0.8993
Epoch 66/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1894 - accuracy: 0.9428 - val_loss: 0.3787 - val_accuracy: 0.9032
Epoch 67/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1907 - accuracy: 0.9416 - val_loss: 0.3968 - val_accuracy: 0.9058
Epoch 68/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1898 - accuracy: 0.9419 - val_loss: 0.3894 - val_accuracy: 0.9049
Epoch 69/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1894 - accuracy: 0.9429 - val_loss: 0.3972 - val_accuracy: 0.9077
Epoch 70/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1841 - accuracy: 0.9441 - val_loss: 0.4008 - val_accuracy: 0.9066
Epoch 71/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1862 - accuracy: 0.9436 - val_loss: 0.3914 - val_accuracy: 0.9016
Epoch 72/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1854 - accuracy: 0.9438 - val_loss: 0.3852 - val_accuracy: 0.9052
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1876 - accuracy: 0.9437 - val_loss: 0.4067 - val_accuracy: 0.8986
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1824 - accuracy: 0.9449 - val_loss: 0.3860 - val_accuracy: 0.9027
Epoch 75/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1806 - accuracy: 0.9454 - val_loss: 0.3826 - val_accuracy: 0.9055
Epoch 76/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1823 - accuracy: 0.9449 - val_loss: 0.4141 - val_accuracy: 0.9024
Epoch 77/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1797 - accuracy: 0.9455 - val_loss: 0.4142 - val_accuracy: 0.9044
Epoch 78/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1822 - accuracy: 0.9450 - val_loss: 0.3960 - val_accuracy: 0.9034
Epoch 79/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1782 - accuracy: 0.9457 - val_loss: 0.3877 - val_accuracy: 0.9059
Epoch 80/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1759 - accuracy: 0.9464 - val_loss: 0.4078 - val_accuracy: 0.9089
Epoch 81/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1767 - accuracy: 0.9464 - val_loss: 0.4948 - val_accuracy: 0.9019
Epoch 82/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1764 - accuracy: 0.9463 - val_loss: 0.4341 - val_accuracy: 0.9014
Epoch 83/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1731 - accuracy: 0.9475 - val_loss: 0.4258 - val_accuracy: 0.9073
Epoch 84/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1758 - accuracy: 0.9469 - val_loss: 0.4076 - val_accuracy: 0.9066
Epoch 85/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1722 - accuracy: 0.9479 - val_loss: 0.4274 - val_accuracy: 0.8984
Epoch 86/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1704 - accuracy: 0.9469 - val_loss: 0.4200 - val_accuracy: 0.8996
Epoch 87/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1704 - accuracy: 0.9481 - val_loss: 0.4492 - val_accuracy: 0.9013
Epoch 88/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1726 - accuracy: 0.9473 - val_loss: 0.4176 - val_accuracy: 0.9055
Epoch 89/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1698 - accuracy: 0.9475 - val_loss: 0.4223 - val_accuracy: 0.9014
Epoch 90/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1665 - accuracy: 0.9490 - val_loss: 0.4423 - val_accuracy: 0.9005
Epoch 91/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1710 - accuracy: 0.9482 - val_loss: 0.4152 - val_accuracy: 0.9035
Epoch 92/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1675 - accuracy: 0.9495 - val_loss: 0.4188 - val_accuracy: 0.9064
Epoch 93/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1649 - accuracy: 0.9501 - val_loss: 0.4207 - val_accuracy: 0.9067
Epoch 94/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1670 - accuracy: 0.9489 - val_loss: 0.5021 - val_accuracy: 0.9038
Epoch 95/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1670 - accuracy: 0.9493 - val_loss: 0.4550 - val_accuracy: 0.9082
Epoch 96/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1653 - accuracy: 0.9489 - val_loss: 0.4542 - val_accuracy: 0.8912
Epoch 97/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1630 - accuracy: 0.9514 - val_loss: 0.4613 - val_accuracy: 0.9062
Epoch 98/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1629 - accuracy: 0.9508 - val_loss: 0.4438 - val_accuracy: 0.9012
Epoch 99/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1655 - accuracy: 0.9502 - val_loss: 0.4202 - val_accuracy: 0.9043
Epoch 100/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1612 - accuracy: 0.9511 - val_loss: 0.4905 - val_accuracy: 0.8998
> accuaracy of test set 88.975
```

In [ ]:

```
summarize_outcome(history = history)
```



In [ ]:

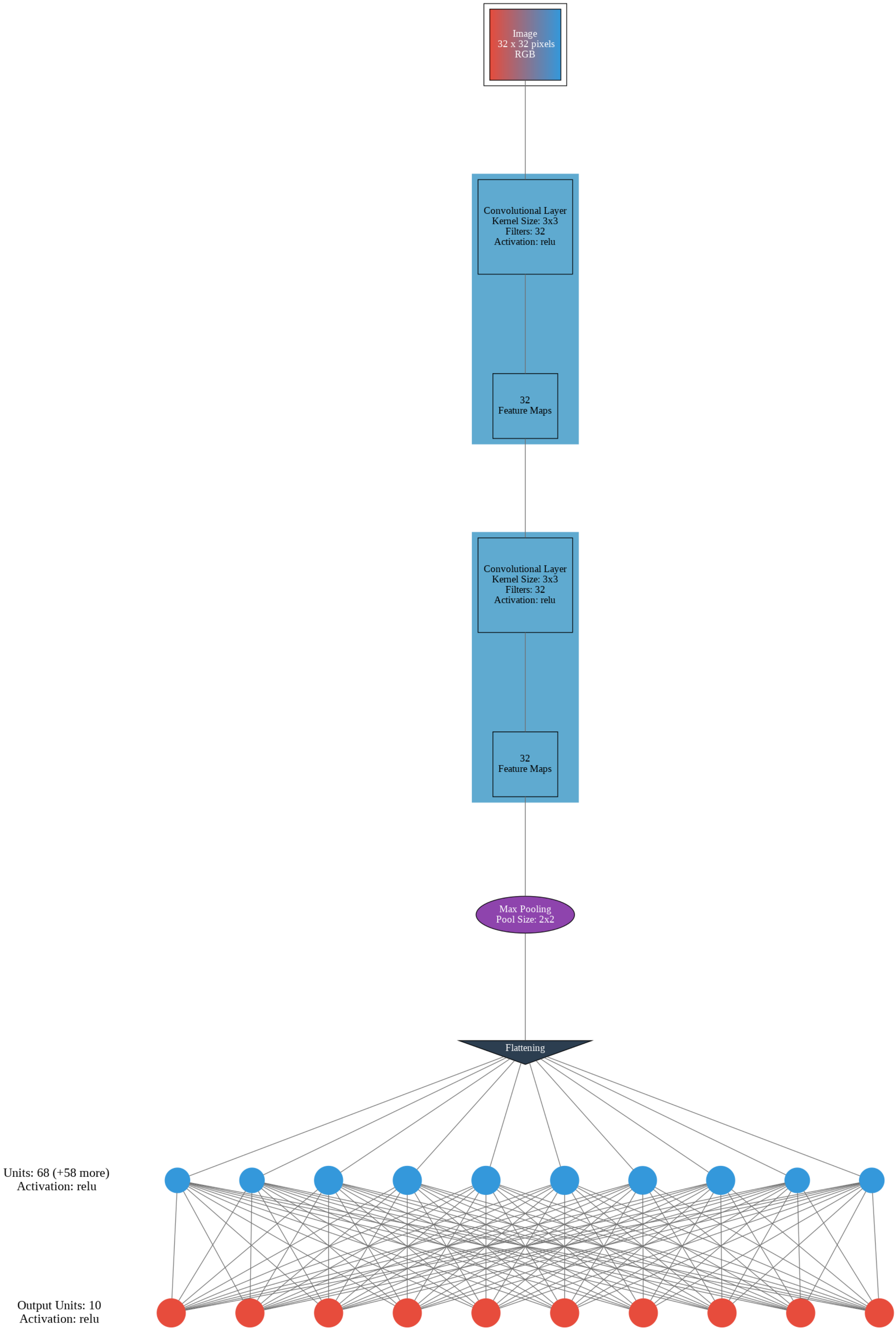
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 32, 32, 32)        896

 conv2d_3 (Conv2D)           (None, 32, 32, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 8192)              0

 dense_2 (Dense)             (None, 68)                557124

 dense_3 (Dense)             (None, 10)                690

=================================================================
Total params: 567,958
Trainable params: 567,958
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
visualizer(model,format='png',view=True, filename='model2')
Image(filename='model2.png')
```

Out[ ]:



In [ ]:

```
#model 3

model2 = Sequential()
model2.add(Conv2D(32,(3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model2.add(MaxPooling2D((2, 2)))
```

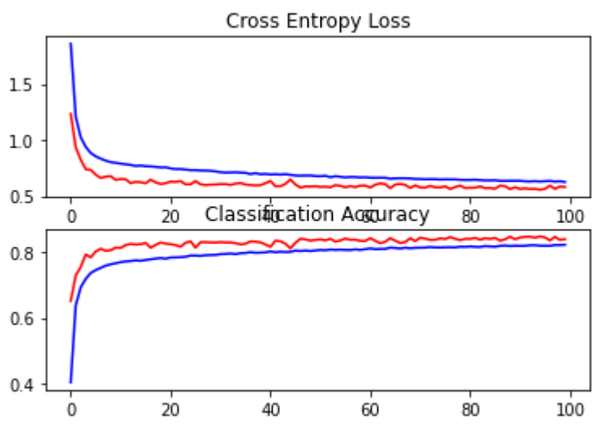
```
model2.add(Flatten())
model2.add(Dense(10, activation='softmax'))
# compile model
model2.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model2.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model2.evaluate(test_imgs, test_labels, verbose=0)
print('> %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 35s 72ms/step - loss: 1.8596 - accuracy: 0.4035 - val_loss: 1.2328 - val_accuracy: 0.6521
Epoch 2/100
487/487 [==============================] - 35s 71ms/step - loss: 1.2075 - accuracy: 0.6384 - val_loss: 0.9354 - val_accuracy: 0.7291
Epoch 3/100
487/487 [==============================] - 35s 72ms/step - loss: 1.0243 - accuracy: 0.6934 - val_loss: 0.8244 - val_accuracy: 0.7551
Epoch 4/100
487/487 [==============================] - 35s 71ms/step - loss: 0.9397 - accuracy: 0.7189 - val_loss: 0.7394 - val_accuracy: 0.7932
Epoch 5/100
487/487 [==============================] - 34s 71ms/step - loss: 0.8844 - accuracy: 0.7369 - val_loss: 0.7343 - val_accuracy: 0.7846
Epoch 6/100
487/487 [==============================] - 34s 70ms/step - loss: 0.8549 - accuracy: 0.7457 - val_loss: 0.6916 - val_accuracy: 0.8021
Epoch 7/100
487/487 [==============================] - 34s 70ms/step - loss: 0.8340 - accuracy: 0.7525 - val_loss: 0.6613 - val_accuracy: 0.8108
Epoch 8/100
487/487 [==============================] - 35s 71ms/step - loss: 0.8164 - accuracy: 0.7593 - val_loss: 0.6734 - val_accuracy: 0.8046
Epoch 9/100
487/487 [==============================] - 34s 70ms/step - loss: 0.8033 - accuracy: 0.7633 - val_loss: 0.6776 - val_accuracy: 0.8062
Epoch 10/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7965 - accuracy: 0.7665 - val_loss: 0.6441 - val_accuracy: 0.8142
Epoch 11/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7900 - accuracy: 0.7699 - val_loss: 0.6526 - val_accuracy: 0.8133
Epoch 12/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7838 - accuracy: 0.7722 - val_loss: 0.6497 - val_accuracy: 0.8220
Epoch 13/100
487/487 [==============================] - 35s 72ms/step - loss: 0.7783 - accuracy: 0.7733 - val_loss: 0.6153 - val_accuracy: 0.8254
Epoch 14/100
487/487 [==============================] - 35s 73ms/step - loss: 0.7683 - accuracy: 0.7753 - val_loss: 0.6271 - val_accuracy: 0.8236
Epoch 15/100
487/487 [==============================] - 35s 73ms/step - loss: 0.7725 - accuracy: 0.7742 - val_loss: 0.6235 - val_accuracy: 0.8247
Epoch 16/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7665 - accuracy: 0.7764 - val_loss: 0.6145 - val_accuracy: 0.8286
Epoch 17/100
487/487 [==============================] - 35s 72ms/step - loss: 0.7645 - accuracy: 0.7786 - val_loss: 0.6466 - val_accuracy: 0.8139
Epoch 18/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7610 - accuracy: 0.7806 - val_loss: 0.6227 - val_accuracy: 0.8225
Epoch 19/100
487/487 [==============================] - 35s 71ms/step - loss: 0.7558 - accuracy: 0.7825 - val_loss: 0.6073 - val_accuracy: 0.8296
Epoch 20/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7572 - accuracy: 0.7805 - val_loss: 0.6148 - val_accuracy: 0.8265
Epoch 21/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7470 - accuracy: 0.7841 - val_loss: 0.6294 - val_accuracy: 0.8246
Epoch 22/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7412 - accuracy: 0.7844 - val_loss: 0.6263 - val_accuracy: 0.8220
Epoch 23/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7416 - accuracy: 0.7850 - val_loss: 0.6337 - val_accuracy: 0.8167
Epoch 24/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7367 - accuracy: 0.7865 - val_loss: 0.6047 - val_accuracy: 0.8289
Epoch 25/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7312 - accuracy: 0.7899 - val_loss: 0.6051 - val_accuracy: 0.8335
Epoch 26/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7317 - accuracy: 0.7899 - val_loss: 0.6340 - val_accuracy: 0.8140
Epoch 27/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7278 - accuracy: 0.7890 - val_loss: 0.6050 - val_accuracy: 0.8308
Epoch 28/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7280 - accuracy: 0.7910 - val_loss: 0.5989 - val_accuracy: 0.8307
Epoch 29/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7252 - accuracy: 0.7916 - val_loss: 0.6021 - val_accuracy: 0.8300
Epoch 30/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7204 - accuracy: 0.7920 - val_loss: 0.6032 - val_accuracy: 0.8296
Epoch 31/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7133 - accuracy: 0.7944 - val_loss: 0.6074 - val_accuracy: 0.8308
Epoch 32/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7094 - accuracy: 0.7949 - val_loss: 0.6071 - val_accuracy: 0.8299
Epoch 33/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7109 - accuracy: 0.7962 - val_loss: 0.5996 - val_accuracy: 0.8301
Epoch 34/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7122 - accuracy: 0.7943 - val_loss: 0.6111 - val_accuracy: 0.8277
Epoch 35/100
487/487 [==============================] - 34s 70ms/step - loss: 0.7094 - accuracy: 0.7972 - val_loss: 0.6164 - val_accuracy: 0.8242
Epoch 36/100
487/487 [==============================] - 34s 69ms/step - loss: 0.7073 - accuracy: 0.7980 - val_loss: 0.6033 - val_accuracy: 0.8260
Epoch 37/100
487/487 [==============================] - 34s 69ms/step - loss: 0.6964 - accuracy: 0.8003 - val_loss: 0.5983 - val_accuracy: 0.8331
Epoch 38/100
487/487 [==============================] - 35s 71ms/step - loss: 0.7050 - accuracy: 0.7984 - val_loss: 0.5955 - val_accuracy: 0.8322
Epoch 39/100
487/487 [==============================] - 35s 72ms/step - loss: 0.6949 - accuracy: 0.7989 - val_loss: 0.5973 - val_accuracy: 0.8302
Epoch 40/100
487/487 [==============================] - 34s 69ms/step - loss: 0.6981 - accuracy: 0.7997 - val_loss: 0.6150 - val_accuracy: 0.8238
Epoch 41/100
487/487 [==============================] - 34s 69ms/step - loss: 0.6926 - accuracy: 0.8024 - val_loss: 0.6355 - val_accuracy: 0.8165
Epoch 42/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6948 - accuracy: 0.8000 - val_loss: 0.5875 - val_accuracy: 0.8355
Epoch 43/100
487/487 [==============================] - 34s 71ms/step - loss: 0.6918 - accuracy: 0.8017 - val_loss: 0.5878 - val_accuracy: 0.8335
Epoch 44/100
487/487 [==============================] - 35s 72ms/step - loss: 0.6954 - accuracy: 0.8007 - val_loss: 0.6085 - val_accuracy: 0.8262
Epoch 45/100
487/487 [==============================] - 35s 71ms/step - loss: 0.6896 - accuracy: 0.8013 - val_loss: 0.6485 - val_accuracy: 0.8119
Epoch 46/100
487/487 [==============================] - 35s 71ms/step - loss: 0.6838 - accuracy: 0.8051 - val_loss: 0.6034 - val_accuracy: 0.8288
Epoch 47/100
487/487 [==============================] - 35s 72ms/step - loss: 0.6819 - accuracy: 0.8038 - val_loss: 0.5756 - val_accuracy: 0.8411
Epoch 48/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6834 - accuracy: 0.8052 - val_loss: 0.5860 - val_accuracy: 0.8394
Epoch 49/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6835 - accuracy: 0.8038 - val_loss: 0.5851 - val_accuracy: 0.8354
Epoch 50/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6786 - accuracy: 0.8056 - val_loss: 0.5880 - val_accuracy: 0.8376
Epoch 51/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6771 - accuracy: 0.8071 - val_loss: 0.5817 - val_accuracy: 0.8390
Epoch 52/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6810 - accuracy: 0.8061 - val_loss: 0.5836 - val_accuracy: 0.8365
Epoch 53/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6690 - accuracy: 0.8084 - val_loss: 0.5787 - val_accuracy: 0.8409
Epoch 54/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6770 - accuracy: 0.8074 - val_loss: 0.5973 - val_accuracy: 0.8326
Epoch 55/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6726 - accuracy: 0.8080 - val_loss: 0.5917 - val_accuracy: 0.8358
Epoch 56/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6668 - accuracy: 0.8074 - val_loss: 0.5786 - val_accuracy: 0.8418
Epoch 57/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6707 - accuracy: 0.8069 - val_loss: 0.5889 - val_accuracy: 0.8380
Epoch 58/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6699 - accuracy: 0.8090 - val_loss: 0.5804 - val_accuracy: 0.8382
Epoch 59/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6664 - accuracy: 0.8073 - val_loss: 0.5932 - val_accuracy: 0.8348
Epoch 60/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6678 - accuracy: 0.8094 - val_loss: 0.5934 - val_accuracy: 0.8345
Epoch 61/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6641 - accuracy: 0.8117 - val_loss: 0.5761 - val_accuracy: 0.8429
Epoch 62/100
```

```
487/487 [==============================] - 33s 67ms/step - loss: 0.6633 - accuracy: 0.8095 - val_loss: 0.6016 - val_accuracy: 0.8350
Epoch 63/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6640 - accuracy: 0.8101 - val_loss: 0.6136 - val_accuracy: 0.8275
Epoch 64/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6637 - accuracy: 0.8103 - val_loss: 0.6066 - val_accuracy: 0.8320
Epoch 65/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6549 - accuracy: 0.8142 - val_loss: 0.5702 - val_accuracy: 0.8430
Epoch 66/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6583 - accuracy: 0.8130 - val_loss: 0.6029 - val_accuracy: 0.8353
Epoch 67/100
487/487 [==============================] - 33s 69ms/step - loss: 0.6589 - accuracy: 0.8114 - val_loss: 0.6084 - val_accuracy: 0.8281
Epoch 68/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6588 - accuracy: 0.8139 - val_loss: 0.5999 - val_accuracy: 0.8311
Epoch 69/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6569 - accuracy: 0.8123 - val_loss: 0.5724 - val_accuracy: 0.8428
Epoch 70/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6530 - accuracy: 0.8136 - val_loss: 0.5954 - val_accuracy: 0.8340
Epoch 71/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6522 - accuracy: 0.8139 - val_loss: 0.5769 - val_accuracy: 0.8443
Epoch 72/100
487/487 [==============================] - 33s 69ms/step - loss: 0.6502 - accuracy: 0.8144 - val_loss: 0.5759 - val_accuracy: 0.8417
Epoch 73/100
487/487 [==============================] - 33s 69ms/step - loss: 0.6506 - accuracy: 0.8159 - val_loss: 0.5904 - val_accuracy: 0.8377
Epoch 74/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6505 - accuracy: 0.8147 - val_loss: 0.5783 - val_accuracy: 0.8380
Epoch 75/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6479 - accuracy: 0.8145 - val_loss: 0.5769 - val_accuracy: 0.8423
Epoch 76/100
487/487 [==============================] - 32s 66ms/step - loss: 0.6491 - accuracy: 0.8149 - val_loss: 0.5871 - val_accuracy: 0.8404
Epoch 77/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6487 - accuracy: 0.8153 - val_loss: 0.5631 - val_accuracy: 0.8463
Epoch 78/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6488 - accuracy: 0.8147 - val_loss: 0.5805 - val_accuracy: 0.8411
Epoch 79/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6453 - accuracy: 0.8170 - val_loss: 0.5936 - val_accuracy: 0.8370
Epoch 80/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6441 - accuracy: 0.8165 - val_loss: 0.5720 - val_accuracy: 0.8408
Epoch 81/100
487/487 [==============================] - 34s 70ms/step - loss: 0.6443 - accuracy: 0.8179 - val_loss: 0.5742 - val_accuracy: 0.8407
Epoch 82/100
487/487 [==============================] - 32s 67ms/step - loss: 0.6460 - accuracy: 0.8165 - val_loss: 0.5747 - val_accuracy: 0.8438
Epoch 83/100
487/487 [==============================] - 32s 67ms/step - loss: 0.6427 - accuracy: 0.8163 - val_loss: 0.5857 - val_accuracy: 0.8385
Epoch 84/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6397 - accuracy: 0.8188 - val_loss: 0.5713 - val_accuracy: 0.8427
Epoch 85/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6399 - accuracy: 0.8178 - val_loss: 0.5721 - val_accuracy: 0.8408
Epoch 86/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6412 - accuracy: 0.8170 - val_loss: 0.5657 - val_accuracy: 0.8439
Epoch 87/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6361 - accuracy: 0.8196 - val_loss: 0.5932 - val_accuracy: 0.8353
Epoch 88/100
487/487 [==============================] - 33s 68ms/step - loss: 0.6355 - accuracy: 0.8205 - val_loss: 0.5882 - val_accuracy: 0.8395
Epoch 89/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6364 - accuracy: 0.8190 - val_loss: 0.5619 - val_accuracy: 0.8477
Epoch 90/100
487/487 [==============================] - 32s 67ms/step - loss: 0.6365 - accuracy: 0.8198 - val_loss: 0.5784 - val_accuracy: 0.8397
Epoch 91/100
487/487 [==============================] - 32s 67ms/step - loss: 0.6362 - accuracy: 0.8195 - val_loss: 0.5635 - val_accuracy: 0.8449
Epoch 92/100
487/487 [==============================] - 32s 67ms/step - loss: 0.6320 - accuracy: 0.8197 - val_loss: 0.5686 - val_accuracy: 0.8471
Epoch 93/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6306 - accuracy: 0.8214 - val_loss: 0.5635 - val_accuracy: 0.8475
Epoch 94/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6331 - accuracy: 0.8215 - val_loss: 0.5643 - val_accuracy: 0.8446
Epoch 95/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6303 - accuracy: 0.8200 - val_loss: 0.5569 - val_accuracy: 0.8484
Epoch 96/100
487/487 [==============================] - 32s 66ms/step - loss: 0.6329 - accuracy: 0.8198 - val_loss: 0.5647 - val_accuracy: 0.8453
Epoch 97/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6353 - accuracy: 0.8198 - val_loss: 0.5933 - val_accuracy: 0.8360
Epoch 98/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6294 - accuracy: 0.8222 - val_loss: 0.5643 - val_accuracy: 0.8470
Epoch 99/100
487/487 [==============================] - 33s 67ms/step - loss: 0.6323 - accuracy: 0.8220 - val_loss: 0.5852 - val_accuracy: 0.8377
Epoch 100/100
487/487 [==============================] - 32s 67ms/step - loss: 0.6258 - accuracy: 0.8230 - val_loss: 0.5809 - val_accuracy: 0.8401
> 82.522
```

In [ ]:

```
summarize_outcome(history = history)
```



In [ ]:

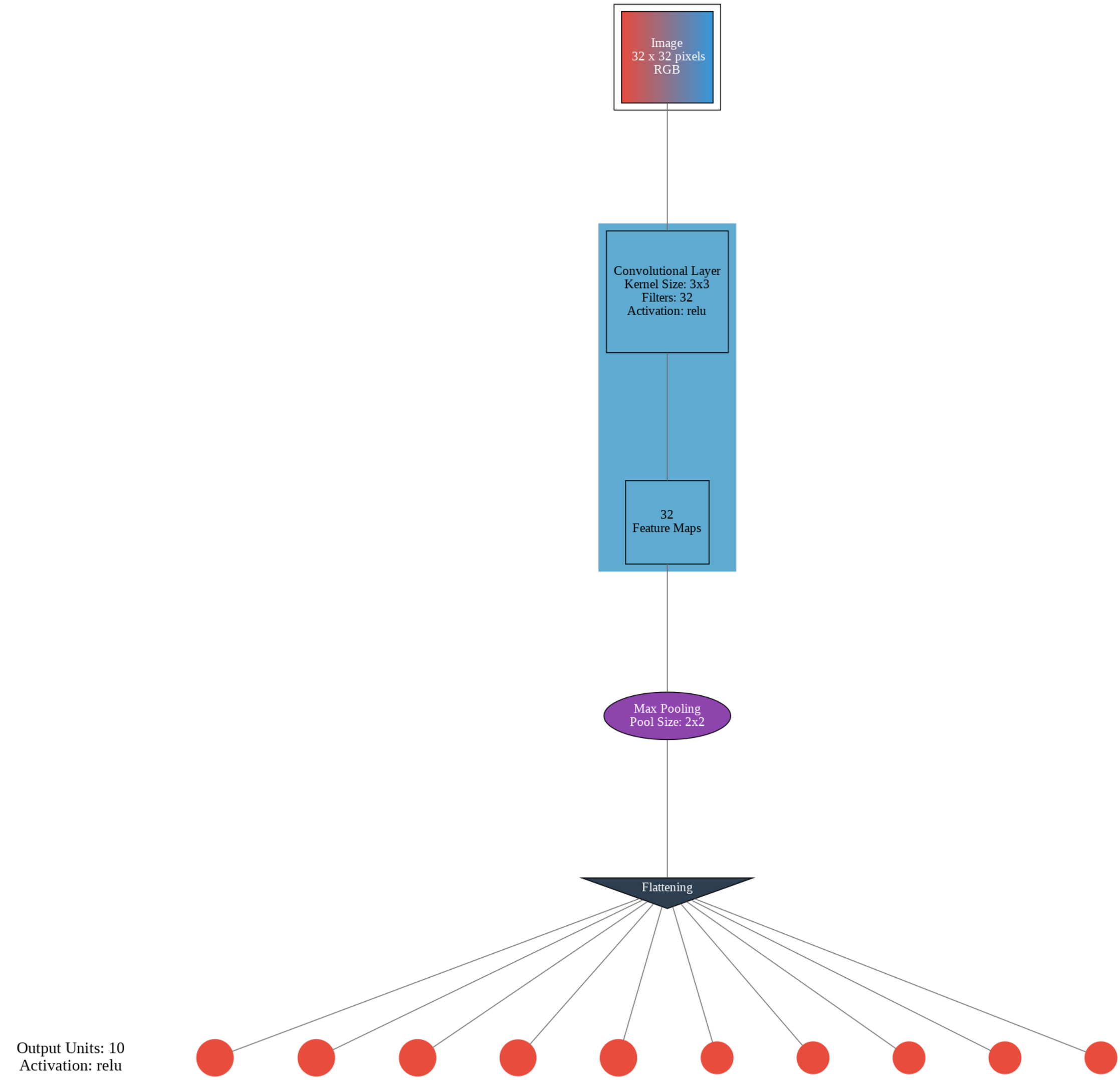
```
model2.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_10 (Conv2D)          (None, 32, 32, 32)        896

 max_pooling2d_6 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 flatten_5 (Flatten)         (None, 8192)              0

 dense_8 (Dense)             (None, 10)                81930

=================================================================
Total params: 82,826
Trainable params: 82,826
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
visualizer(model2,format='png',view=True, filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model3')
Image(filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model3.png')
```

Out[ ]:



In [ ]:

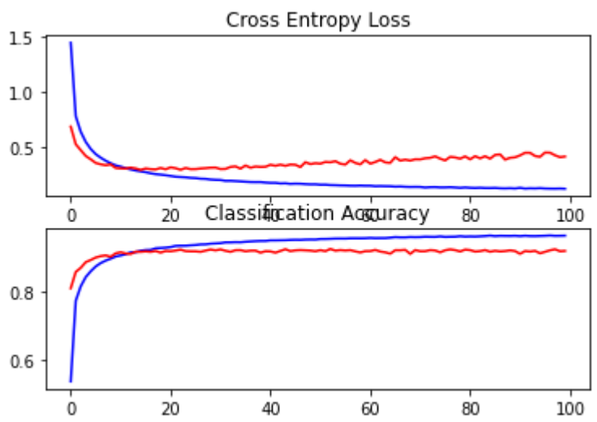
```
#model 4

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_images, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 53ms/step - loss: 1.4409 - accuracy: 0.5350 - val_loss: 0.6819 - val_accuracy: 0.8085
Epoch 2/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7764 - accuracy: 0.7719 - val_loss: 0.5224 - val_accuracy: 0.8565
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6353 - accuracy: 0.8143 - val_loss: 0.4690 - val_accuracy: 0.8686
Epoch 4/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5435 - accuracy: 0.8414 - val_loss: 0.4162 - val_accuracy: 0.8853
Epoch 5/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4796 - accuracy: 0.8586 - val_loss: 0.3859 - val_accuracy: 0.8916
Epoch 6/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4328 - accuracy: 0.8726 - val_loss: 0.3529 - val_accuracy: 0.8989
Epoch 7/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4004 - accuracy: 0.8827 - val_loss: 0.3406 - val_accuracy: 0.9030
Epoch 8/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3730 - accuracy: 0.8898 - val_loss: 0.3322 - val_accuracy: 0.9051
Epoch 9/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3502 - accuracy: 0.8955 - val_loss: 0.3398 - val_accuracy: 0.9010
Epoch 10/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3308 - accuracy: 0.9019 - val_loss: 0.3072 - val_accuracy: 0.9113
Epoch 11/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3197 - accuracy: 0.9049 - val_loss: 0.3041 - val_accuracy: 0.9144
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3053 - accuracy: 0.9087 - val_loss: 0.3027 - val_accuracy: 0.9108
Epoch 13/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2945 - accuracy: 0.9139 - val_loss: 0.3107 - val_accuracy: 0.9093
Epoch 14/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2841 - accuracy: 0.9149 - val_loss: 0.3033 - val_accuracy: 0.9159
Epoch 15/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2769 - accuracy: 0.9187 - val_loss: 0.2899 - val_accuracy: 0.9156
```

```
Epoch 16/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2694 - accuracy: 0.9208 - val_loss: 0.3030 - val_accuracy: 0.9170
Epoch 17/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2594 - accuracy: 0.9214 - val_loss: 0.2968 - val_accuracy: 0.9156
Epoch 18/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2517 - accuracy: 0.9260 - val_loss: 0.2943 - val_accuracy: 0.9181
Epoch 19/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2472 - accuracy: 0.9268 - val_loss: 0.3076 - val_accuracy: 0.9134
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2417 - accuracy: 0.9278 - val_loss: 0.2957 - val_accuracy: 0.9181
Epoch 21/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2355 - accuracy: 0.9296 - val_loss: 0.3133 - val_accuracy: 0.9173
Epoch 22/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2283 - accuracy: 0.9330 - val_loss: 0.3054 - val_accuracy: 0.9194
Epoch 23/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2255 - accuracy: 0.9336 - val_loss: 0.2910 - val_accuracy: 0.9220
Epoch 24/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2227 - accuracy: 0.9335 - val_loss: 0.3074 - val_accuracy: 0.9177
Epoch 25/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2179 - accuracy: 0.9348 - val_loss: 0.2970 - val_accuracy: 0.9172
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2156 - accuracy: 0.9357 - val_loss: 0.2976 - val_accuracy: 0.9171
Epoch 27/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2103 - accuracy: 0.9374 - val_loss: 0.3029 - val_accuracy: 0.9156
Epoch 28/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2088 - accuracy: 0.9377 - val_loss: 0.3059 - val_accuracy: 0.9192
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2028 - accuracy: 0.9395 - val_loss: 0.3102 - val_accuracy: 0.9225
Epoch 30/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1999 - accuracy: 0.9403 - val_loss: 0.3111 - val_accuracy: 0.9200
Epoch 31/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1984 - accuracy: 0.9411 - val_loss: 0.2977 - val_accuracy: 0.9227
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1909 - accuracy: 0.9434 - val_loss: 0.2991 - val_accuracy: 0.9186
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1914 - accuracy: 0.9434 - val_loss: 0.3167 - val_accuracy: 0.9156
Epoch 34/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1888 - accuracy: 0.9440 - val_loss: 0.3218 - val_accuracy: 0.9204
Epoch 35/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1861 - accuracy: 0.9437 - val_loss: 0.3012 - val_accuracy: 0.9206
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1830 - accuracy: 0.9453 - val_loss: 0.3302 - val_accuracy: 0.9172
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1802 - accuracy: 0.9462 - val_loss: 0.3098 - val_accuracy: 0.9189
Epoch 38/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1795 - accuracy: 0.9470 - val_loss: 0.3202 - val_accuracy: 0.9191
Epoch 39/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1797 - accuracy: 0.9476 - val_loss: 0.3168 - val_accuracy: 0.9132
Epoch 40/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1754 - accuracy: 0.9476 - val_loss: 0.3184 - val_accuracy: 0.9174
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1725 - accuracy: 0.9493 - val_loss: 0.3364 - val_accuracy: 0.9168
Epoch 42/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1725 - accuracy: 0.9493 - val_loss: 0.3274 - val_accuracy: 0.9132
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1672 - accuracy: 0.9495 - val_loss: 0.3372 - val_accuracy: 0.9182
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1686 - accuracy: 0.9498 - val_loss: 0.3270 - val_accuracy: 0.9233
Epoch 45/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1637 - accuracy: 0.9506 - val_loss: 0.3387 - val_accuracy: 0.9176
Epoch 46/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1657 - accuracy: 0.9509 - val_loss: 0.3362 - val_accuracy: 0.9199
Epoch 47/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1647 - accuracy: 0.9508 - val_loss: 0.3157 - val_accuracy: 0.9207
Epoch 48/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1616 - accuracy: 0.9516 - val_loss: 0.3599 - val_accuracy: 0.9200
Epoch 49/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1595 - accuracy: 0.9518 - val_loss: 0.3428 - val_accuracy: 0.9192
Epoch 50/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1598 - accuracy: 0.9516 - val_loss: 0.3507 - val_accuracy: 0.9174
Epoch 51/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1561 - accuracy: 0.9533 - val_loss: 0.3467 - val_accuracy: 0.9205
Epoch 52/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1564 - accuracy: 0.9534 - val_loss: 0.3619 - val_accuracy: 0.9176
Epoch 53/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1527 - accuracy: 0.9539 - val_loss: 0.3590 - val_accuracy: 0.9241
Epoch 54/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1516 - accuracy: 0.9546 - val_loss: 0.3690 - val_accuracy: 0.9191
Epoch 55/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1497 - accuracy: 0.9547 - val_loss: 0.3461 - val_accuracy: 0.9207
Epoch 56/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1489 - accuracy: 0.9546 - val_loss: 0.3384 - val_accuracy: 0.9195
Epoch 57/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1471 - accuracy: 0.9554 - val_loss: 0.3771 - val_accuracy: 0.9206
Epoch 58/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1464 - accuracy: 0.9554 - val_loss: 0.3533 - val_accuracy: 0.9169
Epoch 59/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1481 - accuracy: 0.9554 - val_loss: 0.3402 - val_accuracy: 0.9138
Epoch 60/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1464 - accuracy: 0.9556 - val_loss: 0.3788 - val_accuracy: 0.9176
Epoch 61/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1466 - accuracy: 0.9564 - val_loss: 0.3465 - val_accuracy: 0.9177
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1424 - accuracy: 0.9557 - val_loss: 0.3616 - val_accuracy: 0.9212
Epoch 63/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1437 - accuracy: 0.9560 - val_loss: 0.3813 - val_accuracy: 0.9165
Epoch 64/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1424 - accuracy: 0.9559 - val_loss: 0.3581 - val_accuracy: 0.9142
Epoch 65/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1419 - accuracy: 0.9572 - val_loss: 0.3535 - val_accuracy: 0.9101
Epoch 66/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1386 - accuracy: 0.9588 - val_loss: 0.4057 - val_accuracy: 0.9204
Epoch 67/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1400 - accuracy: 0.9581 - val_loss: 0.3740 - val_accuracy: 0.9201
Epoch 68/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1371 - accuracy: 0.9581 - val_loss: 0.3829 - val_accuracy: 0.9215
Epoch 69/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1357 - accuracy: 0.9591 - val_loss: 0.3745 - val_accuracy: 0.9095
Epoch 70/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1363 - accuracy: 0.9587 - val_loss: 0.3861 - val_accuracy: 0.9208
Epoch 71/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1358 - accuracy: 0.9590 - val_loss: 0.3852 - val_accuracy: 0.9173
Epoch 72/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1305 - accuracy: 0.9600 - val_loss: 0.3954 - val_accuracy: 0.9175
Epoch 73/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1337 - accuracy: 0.9596 - val_loss: 0.3994 - val_accuracy: 0.9206
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1343 - accuracy: 0.9595 - val_loss: 0.4126 - val_accuracy: 0.9195
Epoch 75/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1314 - accuracy: 0.9610 - val_loss: 0.3917 - val_accuracy: 0.9142
Epoch 76/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1317 - accuracy: 0.9594 - val_loss: 0.3758 - val_accuracy: 0.9202
Epoch 77/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1335 - accuracy: 0.9599 - val_loss: 0.4079 - val_accuracy: 0.9228
Epoch 78/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1308 - accuracy: 0.9601 - val_loss: 0.4031 - val_accuracy: 0.9206
Epoch 79/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1275 - accuracy: 0.9616 - val_loss: 0.3916 - val_accuracy: 0.9166
Epoch 80/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1310 - accuracy: 0.9604 - val_loss: 0.4123 - val_accuracy: 0.9222
Epoch 81/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1276 - accuracy: 0.9609 - val_loss: 0.3857 - val_accuracy: 0.9206
Epoch 82/100
```

```
487/487 [==============================] - 26s 54ms/step - loss: 0.1260 - accuracy: 0.9605 - val_loss: 0.4144 - val_accuracy: 0.9164
Epoch 83/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1279 - accuracy: 0.9613 - val_loss: 0.3945 - val_accuracy: 0.9176
Epoch 84/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1255 - accuracy: 0.9622 - val_loss: 0.4139 - val_accuracy: 0.9207
Epoch 85/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1252 - accuracy: 0.9635 - val_loss: 0.3917 - val_accuracy: 0.9159
Epoch 86/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1267 - accuracy: 0.9619 - val_loss: 0.4259 - val_accuracy: 0.9186
Epoch 87/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1235 - accuracy: 0.9628 - val_loss: 0.4305 - val_accuracy: 0.9167
Epoch 88/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1224 - accuracy: 0.9626 - val_loss: 0.3828 - val_accuracy: 0.9178
Epoch 89/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1242 - accuracy: 0.9624 - val_loss: 0.3997 - val_accuracy: 0.9199
Epoch 90/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1218 - accuracy: 0.9629 - val_loss: 0.4040 - val_accuracy: 0.9178
Epoch 91/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1277 - accuracy: 0.9618 - val_loss: 0.4216 - val_accuracy: 0.9096
Epoch 92/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1207 - accuracy: 0.9630 - val_loss: 0.4453 - val_accuracy: 0.9179
Epoch 93/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1235 - accuracy: 0.9621 - val_loss: 0.4449 - val_accuracy: 0.9164
Epoch 94/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1222 - accuracy: 0.9619 - val_loss: 0.4161 - val_accuracy: 0.9185
Epoch 95/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1247 - accuracy: 0.9625 - val_loss: 0.4081 - val_accuracy: 0.9112
Epoch 96/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1215 - accuracy: 0.9629 - val_loss: 0.4474 - val_accuracy: 0.9156
Epoch 97/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1207 - accuracy: 0.9638 - val_loss: 0.4458 - val_accuracy: 0.9201
Epoch 98/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1202 - accuracy: 0.9627 - val_loss: 0.4222 - val_accuracy: 0.9232
Epoch 99/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1214 - accuracy: 0.9629 - val_loss: 0.4051 - val_accuracy: 0.9173
Epoch 100/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1194 - accuracy: 0.9632 - val_loss: 0.4110 - val_accuracy: 0.9183
> accuaracy of test set 89.866
```

In [ ]:
```
summarize_outcome(history = history)
```

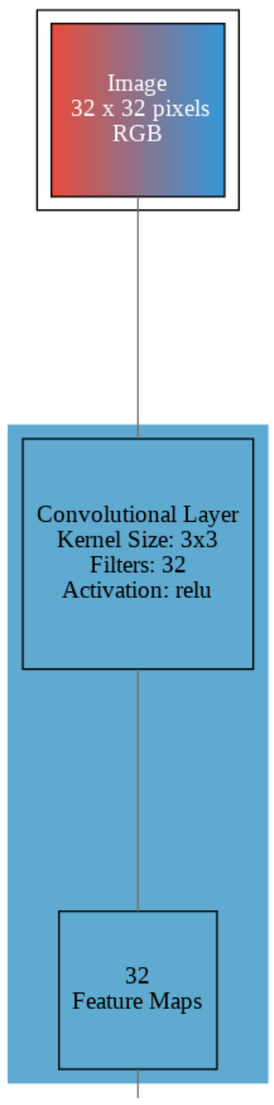


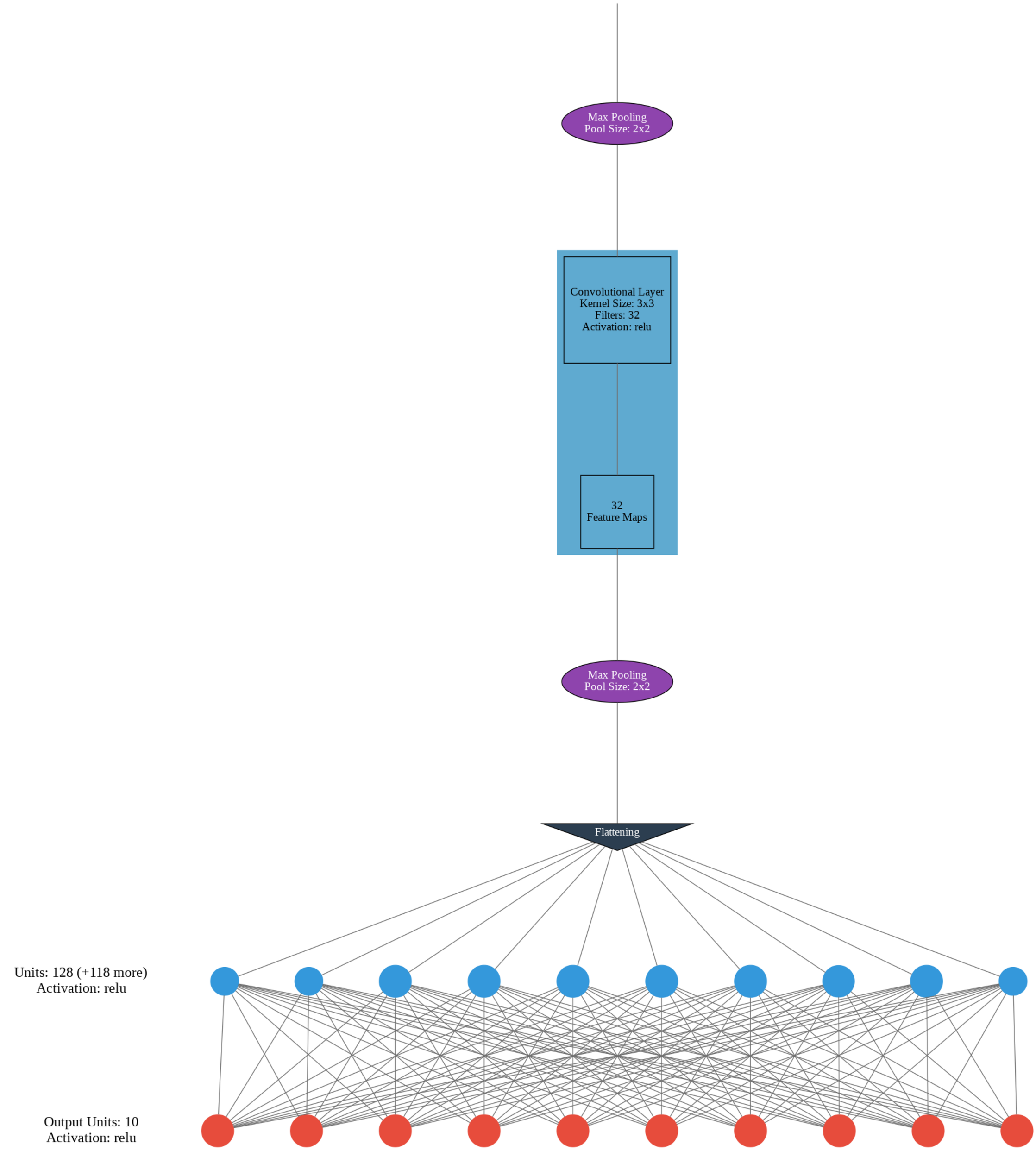
In [ ]:
```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_5 (Conv2D)           (None, 32, 32, 32)        896

 max_pooling2d_3 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 conv2d_6 (Conv2D)           (None, 16, 16, 32)        9248

 max_pooling2d_4 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 flatten_3 (Flatten)         (None, 2048)              0

 dense_5 (Dense)             (None, 128)               262272

 dense_6 (Dense)             (None, 10)                1290

=================================================================
Total params: 273,706
Trainable params: 273,706
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:
```
visualizer(model,format='png',view=True, filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model4')
Image(filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model4.png')
```

Out[ ]:





Final Model with Average Pooling

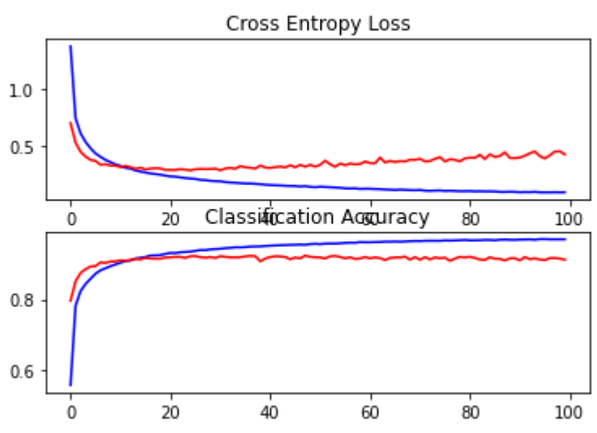
In [ ]:
```
#Final Model with Average Pooling
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(AveragePooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(AveragePooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 53ms/step - loss: 1.3683 - accuracy: 0.5572 - val_loss: 0.7006 - val_accuracy: 0.7959
Epoch 2/100
487/487 [==============================] - 26s 53ms/step - loss: 0.7442 - accuracy: 0.7814 - val_loss: 0.5321 - val_accuracy: 0.8507
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6122 - accuracy: 0.8230 - val_loss: 0.4517 - val_accuracy: 0.8743
Epoch 4/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5375 - accuracy: 0.8431 - val_loss: 0.4094 - val_accuracy: 0.8851
Epoch 5/100
487/487 [==============================] - 26s 53ms/step - loss: 0.4820 - accuracy: 0.8580 - val_loss: 0.3802 - val_accuracy: 0.8927
Epoch 6/100
487/487 [==============================] - 26s 53ms/step - loss: 0.4369 - accuracy: 0.8723 - val_loss: 0.3720 - val_accuracy: 0.8938
Epoch 7/100
487/487 [==============================] - 26s 53ms/step - loss: 0.4073 - accuracy: 0.8819 - val_loss: 0.3380 - val_accuracy: 0.9044
Epoch 8/100
```

```
487/487 [==============================] - 26s 53ms/step - loss: 0.3804 - accuracy: 0.8879 - val_loss: 0.3424 - val_accuracy: 0.9027
Epoch 9/100
487/487 [==============================] - 26s 53ms/step - loss: 0.3609 - accuracy: 0.8936 - val_loss: 0.3319 - val_accuracy: 0.9069
Epoch 10/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3412 - accuracy: 0.8987 - val_loss: 0.3302 - val_accuracy: 0.9075
Epoch 11/100
487/487 [==============================] - 26s 53ms/step - loss: 0.3283 - accuracy: 0.9033 - val_loss: 0.3180 - val_accuracy: 0.9101
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3114 - accuracy: 0.9087 - val_loss: 0.3283 - val_accuracy: 0.9093
Epoch 13/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3052 - accuracy: 0.9096 - val_loss: 0.3171 - val_accuracy: 0.9113
Epoch 14/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2889 - accuracy: 0.9152 - val_loss: 0.3029 - val_accuracy: 0.9145
Epoch 15/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2795 - accuracy: 0.9173 - val_loss: 0.3126 - val_accuracy: 0.9126
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2707 - accuracy: 0.9197 - val_loss: 0.2971 - val_accuracy: 0.9173
Epoch 17/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2627 - accuracy: 0.9238 - val_loss: 0.3049 - val_accuracy: 0.9154
Epoch 18/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2571 - accuracy: 0.9242 - val_loss: 0.3084 - val_accuracy: 0.9149
Epoch 19/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2517 - accuracy: 0.9251 - val_loss: 0.3061 - val_accuracy: 0.9151
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2437 - accuracy: 0.9284 - val_loss: 0.2933 - val_accuracy: 0.9183
Epoch 21/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2369 - accuracy: 0.9311 - val_loss: 0.2937 - val_accuracy: 0.9187
Epoch 22/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2344 - accuracy: 0.9308 - val_loss: 0.2931 - val_accuracy: 0.9199
Epoch 23/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2279 - accuracy: 0.9332 - val_loss: 0.3003 - val_accuracy: 0.9200
Epoch 24/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2228 - accuracy: 0.9339 - val_loss: 0.2962 - val_accuracy: 0.9172
Epoch 25/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2185 - accuracy: 0.9349 - val_loss: 0.2895 - val_accuracy: 0.9218
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2148 - accuracy: 0.9371 - val_loss: 0.2993 - val_accuracy: 0.9224
Epoch 27/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2064 - accuracy: 0.9392 - val_loss: 0.3020 - val_accuracy: 0.9191
Epoch 28/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2047 - accuracy: 0.9391 - val_loss: 0.3008 - val_accuracy: 0.9177
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1986 - accuracy: 0.9412 - val_loss: 0.3009 - val_accuracy: 0.9195
Epoch 30/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1956 - accuracy: 0.9421 - val_loss: 0.3045 - val_accuracy: 0.9171
Epoch 31/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1945 - accuracy: 0.9430 - val_loss: 0.2907 - val_accuracy: 0.9218
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1874 - accuracy: 0.9446 - val_loss: 0.3053 - val_accuracy: 0.9201
Epoch 33/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1838 - accuracy: 0.9455 - val_loss: 0.3113 - val_accuracy: 0.9187
Epoch 34/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1817 - accuracy: 0.9471 - val_loss: 0.3037 - val_accuracy: 0.9187
Epoch 35/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1784 - accuracy: 0.9470 - val_loss: 0.3265 - val_accuracy: 0.9189
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1754 - accuracy: 0.9476 - val_loss: 0.3164 - val_accuracy: 0.9215
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1752 - accuracy: 0.9490 - val_loss: 0.3121 - val_accuracy: 0.9227
Epoch 38/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1722 - accuracy: 0.9495 - val_loss: 0.3049 - val_accuracy: 0.9221
Epoch 39/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1693 - accuracy: 0.9496 - val_loss: 0.3316 - val_accuracy: 0.9075
Epoch 40/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1644 - accuracy: 0.9509 - val_loss: 0.3136 - val_accuracy: 0.9152
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1619 - accuracy: 0.9518 - val_loss: 0.3110 - val_accuracy: 0.9196
Epoch 42/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1598 - accuracy: 0.9524 - val_loss: 0.3167 - val_accuracy: 0.9218
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1582 - accuracy: 0.9530 - val_loss: 0.3230 - val_accuracy: 0.9213
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1555 - accuracy: 0.9533 - val_loss: 0.3148 - val_accuracy: 0.9196
Epoch 45/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1541 - accuracy: 0.9540 - val_loss: 0.3338 - val_accuracy: 0.9134
Epoch 46/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1534 - accuracy: 0.9542 - val_loss: 0.3160 - val_accuracy: 0.9181
Epoch 47/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1492 - accuracy: 0.9548 - val_loss: 0.3374 - val_accuracy: 0.9158
Epoch 48/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1523 - accuracy: 0.9543 - val_loss: 0.3210 - val_accuracy: 0.9236
Epoch 49/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1462 - accuracy: 0.9562 - val_loss: 0.3375 - val_accuracy: 0.9202
Epoch 50/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1427 - accuracy: 0.9570 - val_loss: 0.3218 - val_accuracy: 0.9188
Epoch 51/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1472 - accuracy: 0.9560 - val_loss: 0.3371 - val_accuracy: 0.9184
Epoch 52/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1450 - accuracy: 0.9577 - val_loss: 0.3737 - val_accuracy: 0.9156
Epoch 53/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1424 - accuracy: 0.9576 - val_loss: 0.3426 - val_accuracy: 0.9216
Epoch 54/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1402 - accuracy: 0.9578 - val_loss: 0.3207 - val_accuracy: 0.9226
Epoch 55/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1379 - accuracy: 0.9585 - val_loss: 0.3487 - val_accuracy: 0.9200
Epoch 56/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1331 - accuracy: 0.9596 - val_loss: 0.3346 - val_accuracy: 0.9157
Epoch 57/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1319 - accuracy: 0.9602 - val_loss: 0.3486 - val_accuracy: 0.9188
Epoch 58/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1329 - accuracy: 0.9600 - val_loss: 0.3416 - val_accuracy: 0.9143
Epoch 59/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1280 - accuracy: 0.9619 - val_loss: 0.3444 - val_accuracy: 0.9155
Epoch 60/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1298 - accuracy: 0.9615 - val_loss: 0.3661 - val_accuracy: 0.9196
Epoch 61/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1296 - accuracy: 0.9614 - val_loss: 0.3524 - val_accuracy: 0.9159
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1270 - accuracy: 0.9619 - val_loss: 0.3514 - val_accuracy: 0.9186
Epoch 63/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1261 - accuracy: 0.9620 - val_loss: 0.4018 - val_accuracy: 0.9164
Epoch 64/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1228 - accuracy: 0.9632 - val_loss: 0.3582 - val_accuracy: 0.9109
Epoch 65/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1215 - accuracy: 0.9629 - val_loss: 0.3720 - val_accuracy: 0.9170
Epoch 66/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1186 - accuracy: 0.9635 - val_loss: 0.3618 - val_accuracy: 0.9185
Epoch 67/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1220 - accuracy: 0.9629 - val_loss: 0.3705 - val_accuracy: 0.9185
Epoch 68/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1198 - accuracy: 0.9637 - val_loss: 0.3671 - val_accuracy: 0.9208
Epoch 69/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1173 - accuracy: 0.9644 - val_loss: 0.3819 - val_accuracy: 0.9121
Epoch 70/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1165 - accuracy: 0.9650 - val_loss: 0.3808 - val_accuracy: 0.9194
Epoch 71/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1189 - accuracy: 0.9647 - val_loss: 0.3902 - val_accuracy: 0.9120
Epoch 72/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1128 - accuracy: 0.9655 - val_loss: 0.3672 - val_accuracy: 0.9192
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1116 - accuracy: 0.9656 - val_loss: 0.3707 - val_accuracy: 0.9128
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1134 - accuracy: 0.9657 - val_loss: 0.3904 - val_accuracy: 0.9185
```

```
Epoch 75/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1145 - accuracy: 0.9655 - val_loss: 0.4044 - val_accuracy: 0.9160
Epoch 76/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1106 - accuracy: 0.9667 - val_loss: 0.3661 - val_accuracy: 0.9178
Epoch 77/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1124 - accuracy: 0.9662 - val_loss: 0.3867 - val_accuracy: 0.9089
Epoch 78/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1103 - accuracy: 0.9666 - val_loss: 0.3808 - val_accuracy: 0.9152
Epoch 79/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1082 - accuracy: 0.9671 - val_loss: 0.3674 - val_accuracy: 0.9199
Epoch 80/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1084 - accuracy: 0.9678 - val_loss: 0.3943 - val_accuracy: 0.9186
Epoch 81/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1100 - accuracy: 0.9668 - val_loss: 0.4000 - val_accuracy: 0.9203
Epoch 82/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1065 - accuracy: 0.9670 - val_loss: 0.3990 - val_accuracy: 0.9169
Epoch 83/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1080 - accuracy: 0.9671 - val_loss: 0.4247 - val_accuracy: 0.9126
Epoch 84/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1067 - accuracy: 0.9680 - val_loss: 0.3882 - val_accuracy: 0.9114
Epoch 85/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1071 - accuracy: 0.9678 - val_loss: 0.4293 - val_accuracy: 0.9190
Epoch 86/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1058 - accuracy: 0.9672 - val_loss: 0.4062 - val_accuracy: 0.9148
Epoch 87/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1017 - accuracy: 0.9687 - val_loss: 0.4143 - val_accuracy: 0.9145
Epoch 88/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1018 - accuracy: 0.9690 - val_loss: 0.4470 - val_accuracy: 0.9098
Epoch 89/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1032 - accuracy: 0.9685 - val_loss: 0.3973 - val_accuracy: 0.9157
Epoch 90/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1021 - accuracy: 0.9682 - val_loss: 0.3962 - val_accuracy: 0.9162
Epoch 91/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0992 - accuracy: 0.9690 - val_loss: 0.4005 - val_accuracy: 0.9099
Epoch 92/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1006 - accuracy: 0.9691 - val_loss: 0.4174 - val_accuracy: 0.9196
Epoch 93/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0999 - accuracy: 0.9696 - val_loss: 0.4375 - val_accuracy: 0.9135
Epoch 94/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1025 - accuracy: 0.9685 - val_loss: 0.4548 - val_accuracy: 0.9159
Epoch 95/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0983 - accuracy: 0.9703 - val_loss: 0.4129 - val_accuracy: 0.9125
Epoch 96/100
487/487 [==============================] - 27s 54ms/step - loss: 0.0980 - accuracy: 0.9704 - val_loss: 0.3936 - val_accuracy: 0.9108
Epoch 97/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0978 - accuracy: 0.9697 - val_loss: 0.4199 - val_accuracy: 0.9165
Epoch 98/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0993 - accuracy: 0.9697 - val_loss: 0.4533 - val_accuracy: 0.9166
Epoch 99/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0981 - accuracy: 0.9698 - val_loss: 0.4552 - val_accuracy: 0.9151
Epoch 100/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0988 - accuracy: 0.9697 - val_loss: 0.4280 - val_accuracy: 0.9120
> accuaracy of test set 89.978
```

In [ ]:
```
summarize_outcome(history = history)
```



In [ ]:
```
model.summary()
```

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_7 (Conv2D)           (None, 32, 32, 32)        896

 average_pooling2d (AverageP  (None, 16, 16, 32)       0
 ooling2D)

 conv2d_8 (Conv2D)           (None, 16, 16, 32)        9248

 average_pooling2d_1 (Averag  (None, 8, 8, 32)         0
 ePooling2D)

 flatten_4 (Flatten)         (None, 2048)              0

 dense_7 (Dense)             (None, 128)               262272

 dense_8 (Dense)             (None, 10)                1290

=================================================================
Total params: 273,706
Trainable params: 273,706
Non-trainable params: 0
_________________________________________________________________
```

Final Model with Batch Normalization

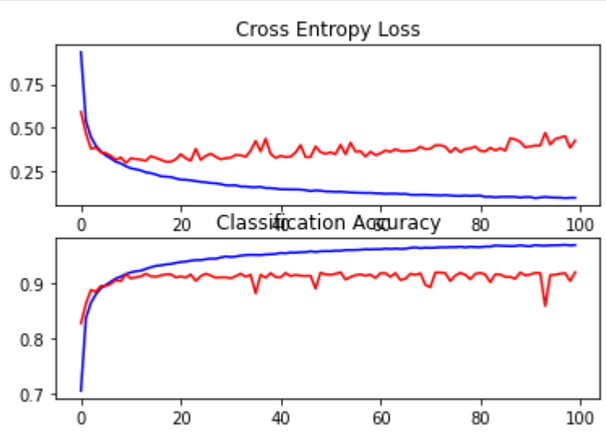
In [ ]:
```
#Final Model with Batch Normalization

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(BatchNormalization())
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 28s 55ms/step - loss: 0.9349 - accuracy: 0.7042 - val_loss: 0.5902 - val_accuracy: 0.8273
Epoch 2/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5379 - accuracy: 0.8370 - val_loss: 0.4667 - val_accuracy: 0.8635
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4455 - accuracy: 0.8650 - val_loss: 0.3771 - val_accuracy: 0.8885
```

```
Epoch 4/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3946 - accuracy: 0.8805 - val_loss: 0.3840 - val_accuracy: 0.8852
Epoch 5/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3615 - accuracy: 0.8917 - val_loss: 0.3555 - val_accuracy: 0.8959
Epoch 6/100
487/487 [==============================] - 27s 54ms/step - loss: 0.3377 - accuracy: 0.8975 - val_loss: 0.3518 - val_accuracy: 0.8955
Epoch 7/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3209 - accuracy: 0.9036 - val_loss: 0.3360 - val_accuracy: 0.9000
Epoch 8/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3018 - accuracy: 0.9093 - val_loss: 0.3135 - val_accuracy: 0.9065
Epoch 9/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2921 - accuracy: 0.9128 - val_loss: 0.3266 - val_accuracy: 0.9046
Epoch 10/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2762 - accuracy: 0.9175 - val_loss: 0.2952 - val_accuracy: 0.9172
Epoch 11/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2641 - accuracy: 0.9206 - val_loss: 0.3222 - val_accuracy: 0.9094
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2586 - accuracy: 0.9225 - val_loss: 0.3174 - val_accuracy: 0.9118
Epoch 13/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2517 - accuracy: 0.9236 - val_loss: 0.3138 - val_accuracy: 0.9132
Epoch 14/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2406 - accuracy: 0.9269 - val_loss: 0.3072 - val_accuracy: 0.9181
Epoch 15/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2360 - accuracy: 0.9298 - val_loss: 0.3342 - val_accuracy: 0.9135
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2285 - accuracy: 0.9325 - val_loss: 0.3245 - val_accuracy: 0.9125
Epoch 17/100
487/487 [==============================] - 27s 55ms/step - loss: 0.2178 - accuracy: 0.9333 - val_loss: 0.3136 - val_accuracy: 0.9150
Epoch 18/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2159 - accuracy: 0.9351 - val_loss: 0.3024 - val_accuracy: 0.9170
Epoch 19/100
487/487 [==============================] - 27s 54ms/step - loss: 0.2134 - accuracy: 0.9358 - val_loss: 0.3019 - val_accuracy: 0.9172
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2057 - accuracy: 0.9378 - val_loss: 0.3161 - val_accuracy: 0.9113
Epoch 21/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1989 - accuracy: 0.9394 - val_loss: 0.3454 - val_accuracy: 0.9132
Epoch 22/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1978 - accuracy: 0.9402 - val_loss: 0.3220 - val_accuracy: 0.9111
Epoch 23/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1945 - accuracy: 0.9419 - val_loss: 0.3076 - val_accuracy: 0.9167
Epoch 24/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1897 - accuracy: 0.9431 - val_loss: 0.3766 - val_accuracy: 0.9049
Epoch 25/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1860 - accuracy: 0.9429 - val_loss: 0.3127 - val_accuracy: 0.9136
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1831 - accuracy: 0.9445 - val_loss: 0.3351 - val_accuracy: 0.9182
Epoch 27/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1788 - accuracy: 0.9456 - val_loss: 0.3489 - val_accuracy: 0.9148
Epoch 28/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1771 - accuracy: 0.9455 - val_loss: 0.3308 - val_accuracy: 0.9108
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1729 - accuracy: 0.9477 - val_loss: 0.3161 - val_accuracy: 0.9115
Epoch 30/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1660 - accuracy: 0.9498 - val_loss: 0.3227 - val_accuracy: 0.9115
Epoch 31/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1655 - accuracy: 0.9486 - val_loss: 0.3249 - val_accuracy: 0.9097
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1659 - accuracy: 0.9495 - val_loss: 0.3412 - val_accuracy: 0.9135
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1587 - accuracy: 0.9512 - val_loss: 0.3393 - val_accuracy: 0.9182
Epoch 34/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1588 - accuracy: 0.9521 - val_loss: 0.3300 - val_accuracy: 0.9126
Epoch 35/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1553 - accuracy: 0.9527 - val_loss: 0.3661 - val_accuracy: 0.9162
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1549 - accuracy: 0.9525 - val_loss: 0.4217 - val_accuracy: 0.8824
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1566 - accuracy: 0.9522 - val_loss: 0.3615 - val_accuracy: 0.9163
Epoch 38/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1501 - accuracy: 0.9533 - val_loss: 0.4354 - val_accuracy: 0.9107
Epoch 39/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1495 - accuracy: 0.9536 - val_loss: 0.3469 - val_accuracy: 0.9192
Epoch 40/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1469 - accuracy: 0.9542 - val_loss: 0.3238 - val_accuracy: 0.9120
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1429 - accuracy: 0.9560 - val_loss: 0.3361 - val_accuracy: 0.9115
Epoch 42/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1427 - accuracy: 0.9554 - val_loss: 0.3288 - val_accuracy: 0.9195
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1419 - accuracy: 0.9569 - val_loss: 0.3312 - val_accuracy: 0.9138
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1419 - accuracy: 0.9566 - val_loss: 0.3559 - val_accuracy: 0.9159
Epoch 45/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1409 - accuracy: 0.9573 - val_loss: 0.3984 - val_accuracy: 0.9145
Epoch 46/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1381 - accuracy: 0.9575 - val_loss: 0.3280 - val_accuracy: 0.9141
Epoch 47/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1323 - accuracy: 0.9591 - val_loss: 0.3293 - val_accuracy: 0.9145
Epoch 48/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1358 - accuracy: 0.9578 - val_loss: 0.3903 - val_accuracy: 0.8910
Epoch 49/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1344 - accuracy: 0.9591 - val_loss: 0.3577 - val_accuracy: 0.9200
Epoch 50/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1301 - accuracy: 0.9593 - val_loss: 0.3475 - val_accuracy: 0.9169
Epoch 51/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1295 - accuracy: 0.9593 - val_loss: 0.3570 - val_accuracy: 0.9162
Epoch 52/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1277 - accuracy: 0.9602 - val_loss: 0.3451 - val_accuracy: 0.9179
Epoch 53/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1285 - accuracy: 0.9600 - val_loss: 0.3999 - val_accuracy: 0.9207
Epoch 54/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1266 - accuracy: 0.9613 - val_loss: 0.3448 - val_accuracy: 0.9075
Epoch 55/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1244 - accuracy: 0.9613 - val_loss: 0.4119 - val_accuracy: 0.9122
Epoch 56/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1231 - accuracy: 0.9615 - val_loss: 0.3608 - val_accuracy: 0.9154
Epoch 57/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1224 - accuracy: 0.9619 - val_loss: 0.3632 - val_accuracy: 0.9168
Epoch 58/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1212 - accuracy: 0.9629 - val_loss: 0.3318 - val_accuracy: 0.9146
Epoch 59/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1218 - accuracy: 0.9624 - val_loss: 0.3605 - val_accuracy: 0.9165
Epoch 60/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1194 - accuracy: 0.9630 - val_loss: 0.3404 - val_accuracy: 0.9103
Epoch 61/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1189 - accuracy: 0.9628 - val_loss: 0.3507 - val_accuracy: 0.9191
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1165 - accuracy: 0.9639 - val_loss: 0.3673 - val_accuracy: 0.9191
Epoch 63/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1174 - accuracy: 0.9632 - val_loss: 0.3596 - val_accuracy: 0.9124
Epoch 64/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1175 - accuracy: 0.9641 - val_loss: 0.3748 - val_accuracy: 0.9194
Epoch 65/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1157 - accuracy: 0.9633 - val_loss: 0.3637 - val_accuracy: 0.9065
Epoch 66/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1163 - accuracy: 0.9638 - val_loss: 0.3632 - val_accuracy: 0.9087
Epoch 67/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1113 - accuracy: 0.9654 - val_loss: 0.3659 - val_accuracy: 0.9180
Epoch 68/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1106 - accuracy: 0.9661 - val_loss: 0.3695 - val_accuracy: 0.9161
Epoch 69/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1110 - accuracy: 0.9647 - val_loss: 0.3894 - val_accuracy: 0.9189
Epoch 70/100
```

```
487/487 [==============================] - 26s 54ms/step - loss: 0.1113 - accuracy: 0.9657 - val_loss: 0.3744 - val_accuracy: 0.8977
Epoch 71/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1097 - accuracy: 0.9653 - val_loss: 0.3773 - val_accuracy: 0.8933
Epoch 72/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1086 - accuracy: 0.9656 - val_loss: 0.3972 - val_accuracy: 0.9203
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1074 - accuracy: 0.9665 - val_loss: 0.3974 - val_accuracy: 0.9204
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1094 - accuracy: 0.9662 - val_loss: 0.3887 - val_accuracy: 0.9191
Epoch 75/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1069 - accuracy: 0.9668 - val_loss: 0.3556 - val_accuracy: 0.9047
Epoch 76/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1047 - accuracy: 0.9667 - val_loss: 0.3819 - val_accuracy: 0.9146
Epoch 77/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1046 - accuracy: 0.9675 - val_loss: 0.3576 - val_accuracy: 0.9198
Epoch 78/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1064 - accuracy: 0.9663 - val_loss: 0.3753 - val_accuracy: 0.9163
Epoch 79/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1047 - accuracy: 0.9675 - val_loss: 0.3788 - val_accuracy: 0.9082
Epoch 80/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1058 - accuracy: 0.9668 - val_loss: 0.3896 - val_accuracy: 0.9179
Epoch 81/100
487/487 [==============================] - 27s 54ms/step - loss: 0.1063 - accuracy: 0.9668 - val_loss: 0.3646 - val_accuracy: 0.9196
Epoch 82/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0989 - accuracy: 0.9677 - val_loss: 0.3620 - val_accuracy: 0.9166
Epoch 83/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0998 - accuracy: 0.9677 - val_loss: 0.3829 - val_accuracy: 0.9024
Epoch 84/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0963 - accuracy: 0.9697 - val_loss: 0.3667 - val_accuracy: 0.9170
Epoch 85/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0988 - accuracy: 0.9691 - val_loss: 0.3802 - val_accuracy: 0.9168
Epoch 86/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0993 - accuracy: 0.9689 - val_loss: 0.3653 - val_accuracy: 0.9122
Epoch 87/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0992 - accuracy: 0.9685 - val_loss: 0.4380 - val_accuracy: 0.9124
Epoch 88/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0989 - accuracy: 0.9687 - val_loss: 0.4311 - val_accuracy: 0.9088
Epoch 89/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0948 - accuracy: 0.9700 - val_loss: 0.4162 - val_accuracy: 0.9205
Epoch 90/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0980 - accuracy: 0.9688 - val_loss: 0.3851 - val_accuracy: 0.9151
Epoch 91/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0978 - accuracy: 0.9682 - val_loss: 0.3902 - val_accuracy: 0.9166
Epoch 92/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0917 - accuracy: 0.9704 - val_loss: 0.3959 - val_accuracy: 0.9199
Epoch 93/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0952 - accuracy: 0.9696 - val_loss: 0.3955 - val_accuracy: 0.9192
Epoch 94/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0988 - accuracy: 0.9693 - val_loss: 0.4690 - val_accuracy: 0.8591
Epoch 95/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0959 - accuracy: 0.9699 - val_loss: 0.4015 - val_accuracy: 0.9155
Epoch 96/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0947 - accuracy: 0.9700 - val_loss: 0.4328 - val_accuracy: 0.9163
Epoch 97/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0943 - accuracy: 0.9702 - val_loss: 0.4416 - val_accuracy: 0.9182
Epoch 98/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0917 - accuracy: 0.9709 - val_loss: 0.4501 - val_accuracy: 0.9192
Epoch 99/100
487/487 [==============================] - 27s 55ms/step - loss: 0.0938 - accuracy: 0.9698 - val_loss: 0.3827 - val_accuracy: 0.9046
Epoch 100/100
487/487 [==============================] - 26s 54ms/step - loss: 0.0936 - accuracy: 0.9705 - val_loss: 0.4243 - val_accuracy: 0.9208
> accuaracy of test set 90.915
```

In [ ]:
```
summarize_outcome(history = history)
```



In [ ]:
```
model.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 32, 32, 32)        896

 batch_normalization (BatchN  (None, 32, 32, 32)       128
 ormalization)

 max_pooling2d_5 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 16, 16, 32)        9248

 batch_normalization_1 (Batc  (None, 16, 16, 32)       128
 hNormalization)

 max_pooling2d_6 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 flatten_5 (Flatten)         (None, 2048)              0

 dense_9 (Dense)             (None, 128)               262272

 batch_normalization_2 (Batc  (None, 128)              512
 hNormalization)

 dense_10 (Dense)            (None, 10)                1290

=================================================================
Total params: 274,474
Trainable params: 274,090
Non-trainable params: 384
_________________________________________________________________
```

Final Model with L2 Normalization

In [ ]:
```
#Final Model with L2 normalization

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', kernel_regularizer = l2(0.001), input_shape=(32, 32, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', kernel_regularizer = l2(0.001)))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform', kernel_regularizer = l2(0.001)))
```

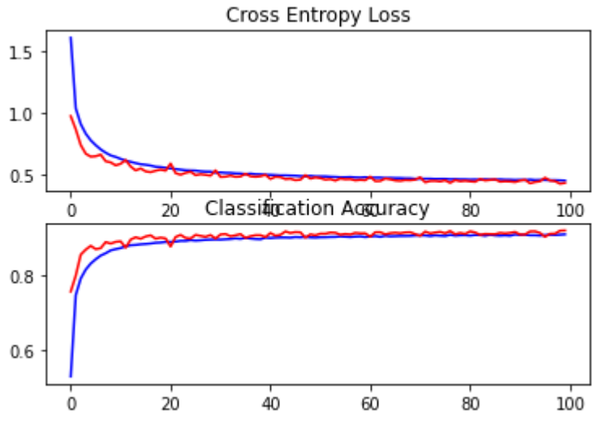
```
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 54ms/step - loss: 1.6125 - accuracy: 0.5295 - val_loss: 0.9744 - val_accuracy: 0.7553
Epoch 2/100
487/487 [==============================] - 26s 53ms/step - loss: 1.0389 - accuracy: 0.7456 - val_loss: 0.8678 - val_accuracy: 0.7974
Epoch 3/100
487/487 [==============================] - 26s 53ms/step - loss: 0.9098 - accuracy: 0.7899 - val_loss: 0.7389 - val_accuracy: 0.8533
Epoch 4/100
487/487 [==============================] - 26s 54ms/step - loss: 0.8320 - accuracy: 0.8136 - val_loss: 0.6688 - val_accuracy: 0.8667
Epoch 5/100
487/487 [==============================] - 26s 53ms/step - loss: 0.7777 - accuracy: 0.8296 - val_loss: 0.6444 - val_accuracy: 0.8768
Epoch 6/100
487/487 [==============================] - 26s 53ms/step - loss: 0.7380 - accuracy: 0.8411 - val_loss: 0.6481 - val_accuracy: 0.8680
Epoch 7/100
487/487 [==============================] - 26s 53ms/step - loss: 0.7035 - accuracy: 0.8509 - val_loss: 0.6612 - val_accuracy: 0.8705
Epoch 8/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6765 - accuracy: 0.8571 - val_loss: 0.6068 - val_accuracy: 0.8868
Epoch 9/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6546 - accuracy: 0.8647 - val_loss: 0.5981 - val_accuracy: 0.8827
Epoch 10/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6421 - accuracy: 0.8680 - val_loss: 0.5731 - val_accuracy: 0.8873
Epoch 11/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6263 - accuracy: 0.8708 - val_loss: 0.5866 - val_accuracy: 0.8887
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6116 - accuracy: 0.8754 - val_loss: 0.6169 - val_accuracy: 0.8718
Epoch 13/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6025 - accuracy: 0.8783 - val_loss: 0.5641 - val_accuracy: 0.8934
Epoch 14/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5921 - accuracy: 0.8792 - val_loss: 0.5323 - val_accuracy: 0.8995
Epoch 15/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5830 - accuracy: 0.8808 - val_loss: 0.5485 - val_accuracy: 0.8955
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5782 - accuracy: 0.8818 - val_loss: 0.5245 - val_accuracy: 0.9016
Epoch 17/100
487/487 [==============================] - 27s 56ms/step - loss: 0.5721 - accuracy: 0.8830 - val_loss: 0.5183 - val_accuracy: 0.9048
Epoch 18/100
487/487 [==============================] - 27s 55ms/step - loss: 0.5622 - accuracy: 0.8847 - val_loss: 0.5301 - val_accuracy: 0.8957
Epoch 19/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5594 - accuracy: 0.8855 - val_loss: 0.5387 - val_accuracy: 0.8985
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5537 - accuracy: 0.8871 - val_loss: 0.5302 - val_accuracy: 0.8964
Epoch 21/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5485 - accuracy: 0.8872 - val_loss: 0.5888 - val_accuracy: 0.8749
Epoch 22/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5438 - accuracy: 0.8880 - val_loss: 0.5100 - val_accuracy: 0.9003
Epoch 23/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5386 - accuracy: 0.8900 - val_loss: 0.4974 - val_accuracy: 0.9059
Epoch 24/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5335 - accuracy: 0.8899 - val_loss: 0.5125 - val_accuracy: 0.8978
Epoch 25/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5322 - accuracy: 0.8919 - val_loss: 0.5221 - val_accuracy: 0.8961
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5309 - accuracy: 0.8906 - val_loss: 0.4936 - val_accuracy: 0.9059
Epoch 27/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5255 - accuracy: 0.8914 - val_loss: 0.5014 - val_accuracy: 0.9033
Epoch 28/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5228 - accuracy: 0.8927 - val_loss: 0.4964 - val_accuracy: 0.9007
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5223 - accuracy: 0.8932 - val_loss: 0.4900 - val_accuracy: 0.9055
Epoch 30/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5145 - accuracy: 0.8937 - val_loss: 0.5330 - val_accuracy: 0.8961
Epoch 31/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5159 - accuracy: 0.8934 - val_loss: 0.4791 - val_accuracy: 0.9070
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5128 - accuracy: 0.8942 - val_loss: 0.4817 - val_accuracy: 0.9082
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5119 - accuracy: 0.8950 - val_loss: 0.4904 - val_accuracy: 0.9022
Epoch 34/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5082 - accuracy: 0.8966 - val_loss: 0.4803 - val_accuracy: 0.9047
Epoch 35/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5072 - accuracy: 0.8954 - val_loss: 0.4829 - val_accuracy: 0.9080
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5027 - accuracy: 0.8979 - val_loss: 0.5050 - val_accuracy: 0.8979
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5062 - accuracy: 0.8962 - val_loss: 0.4847 - val_accuracy: 0.9024
Epoch 38/100
487/487 [==============================] - 27s 55ms/step - loss: 0.5013 - accuracy: 0.8950 - val_loss: 0.4810 - val_accuracy: 0.9050
Epoch 39/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5005 - accuracy: 0.8943 - val_loss: 0.4848 - val_accuracy: 0.9050
Epoch 40/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4973 - accuracy: 0.8985 - val_loss: 0.4965 - val_accuracy: 0.9000
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4960 - accuracy: 0.8974 - val_loss: 0.4627 - val_accuracy: 0.9108
Epoch 42/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4943 - accuracy: 0.8985 - val_loss: 0.4805 - val_accuracy: 0.9038
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4919 - accuracy: 0.8988 - val_loss: 0.4775 - val_accuracy: 0.9060
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4921 - accuracy: 0.8989 - val_loss: 0.4618 - val_accuracy: 0.9150
Epoch 45/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4906 - accuracy: 0.8984 - val_loss: 0.4649 - val_accuracy: 0.9108
Epoch 46/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4895 - accuracy: 0.9000 - val_loss: 0.4529 - val_accuracy: 0.9133
Epoch 47/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4862 - accuracy: 0.8994 - val_loss: 0.4572 - val_accuracy: 0.9126
Epoch 48/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4843 - accuracy: 0.9005 - val_loss: 0.4938 - val_accuracy: 0.8985
Epoch 49/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4876 - accuracy: 0.8992 - val_loss: 0.4655 - val_accuracy: 0.9073
Epoch 50/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4836 - accuracy: 0.8987 - val_loss: 0.4752 - val_accuracy: 0.9047
Epoch 51/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4837 - accuracy: 0.8995 - val_loss: 0.4694 - val_accuracy: 0.9091
Epoch 52/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4834 - accuracy: 0.8997 - val_loss: 0.4576 - val_accuracy: 0.9090
Epoch 53/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4806 - accuracy: 0.9003 - val_loss: 0.4617 - val_accuracy: 0.9120
Epoch 54/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4791 - accuracy: 0.9005 - val_loss: 0.4492 - val_accuracy: 0.9115
Epoch 55/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4766 - accuracy: 0.9019 - val_loss: 0.4662 - val_accuracy: 0.9065
Epoch 56/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4766 - accuracy: 0.9011 - val_loss: 0.4604 - val_accuracy: 0.9083
Epoch 57/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4782 - accuracy: 0.9003 - val_loss: 0.4617 - val_accuracy: 0.9069
Epoch 58/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4751 - accuracy: 0.9014 - val_loss: 0.4513 - val_accuracy: 0.9111
Epoch 59/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4739 - accuracy: 0.9019 - val_loss: 0.4556 - val_accuracy: 0.9098
Epoch 60/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4770 - accuracy: 0.9002 - val_loss: 0.4526 - val_accuracy: 0.9099
Epoch 61/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4735 - accuracy: 0.9016 - val_loss: 0.4818 - val_accuracy: 0.9012
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4735 - accuracy: 0.9028 - val_loss: 0.4451 - val_accuracy: 0.9128
```

```
Epoch 63/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4737 - accuracy: 0.9006 - val_loss: 0.4464 - val_accuracy: 0.9129
Epoch 64/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4702 - accuracy: 0.9034 - val_loss: 0.4664 - val_accuracy: 0.9075
Epoch 65/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4715 - accuracy: 0.9020 - val_loss: 0.4595 - val_accuracy: 0.9104
Epoch 66/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4701 - accuracy: 0.9025 - val_loss: 0.4503 - val_accuracy: 0.9115
Epoch 67/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4697 - accuracy: 0.9030 - val_loss: 0.4472 - val_accuracy: 0.9102
Epoch 68/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4681 - accuracy: 0.9031 - val_loss: 0.4539 - val_accuracy: 0.9123
Epoch 69/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4678 - accuracy: 0.9033 - val_loss: 0.4522 - val_accuracy: 0.9128
Epoch 70/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4641 - accuracy: 0.9042 - val_loss: 0.4572 - val_accuracy: 0.9079
Epoch 71/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4652 - accuracy: 0.9045 - val_loss: 0.4763 - val_accuracy: 0.9033
Epoch 72/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4647 - accuracy: 0.9043 - val_loss: 0.4355 - val_accuracy: 0.9146
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4648 - accuracy: 0.9027 - val_loss: 0.4465 - val_accuracy: 0.9096
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4627 - accuracy: 0.9046 - val_loss: 0.4473 - val_accuracy: 0.9121
Epoch 75/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4633 - accuracy: 0.9037 - val_loss: 0.4427 - val_accuracy: 0.9139
Epoch 76/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4634 - accuracy: 0.9029 - val_loss: 0.4520 - val_accuracy: 0.9099
Epoch 77/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4605 - accuracy: 0.9061 - val_loss: 0.4309 - val_accuracy: 0.9161
Epoch 78/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4613 - accuracy: 0.9038 - val_loss: 0.4539 - val_accuracy: 0.9075
Epoch 79/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4597 - accuracy: 0.9048 - val_loss: 0.4442 - val_accuracy: 0.9110
Epoch 80/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4595 - accuracy: 0.9050 - val_loss: 0.4482 - val_accuracy: 0.9084
Epoch 81/100
487/487 [==============================] - 27s 56ms/step - loss: 0.4578 - accuracy: 0.9040 - val_loss: 0.4437 - val_accuracy: 0.9153
Epoch 82/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4589 - accuracy: 0.9057 - val_loss: 0.4398 - val_accuracy: 0.9123
Epoch 83/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4561 - accuracy: 0.9053 - val_loss: 0.4625 - val_accuracy: 0.9060
Epoch 84/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4601 - accuracy: 0.9035 - val_loss: 0.4506 - val_accuracy: 0.9084
Epoch 85/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4575 - accuracy: 0.9063 - val_loss: 0.4551 - val_accuracy: 0.9079
Epoch 86/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4569 - accuracy: 0.9044 - val_loss: 0.4595 - val_accuracy: 0.9076
Epoch 87/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4596 - accuracy: 0.9038 - val_loss: 0.4407 - val_accuracy: 0.9111
Epoch 88/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4570 - accuracy: 0.9046 - val_loss: 0.4441 - val_accuracy: 0.9097
Epoch 89/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4567 - accuracy: 0.9036 - val_loss: 0.4423 - val_accuracy: 0.9111
Epoch 90/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4549 - accuracy: 0.9070 - val_loss: 0.4377 - val_accuracy: 0.9135
Epoch 91/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4561 - accuracy: 0.9056 - val_loss: 0.4491 - val_accuracy: 0.9068
Epoch 92/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4539 - accuracy: 0.9062 - val_loss: 0.4569 - val_accuracy: 0.9069
Epoch 93/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4579 - accuracy: 0.9048 - val_loss: 0.4272 - val_accuracy: 0.9151
Epoch 94/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4541 - accuracy: 0.9049 - val_loss: 0.4381 - val_accuracy: 0.9154
Epoch 95/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4553 - accuracy: 0.9040 - val_loss: 0.4438 - val_accuracy: 0.9110
Epoch 96/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4543 - accuracy: 0.9051 - val_loss: 0.4729 - val_accuracy: 0.9008
Epoch 97/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4525 - accuracy: 0.9063 - val_loss: 0.4497 - val_accuracy: 0.9087
Epoch 98/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4528 - accuracy: 0.9057 - val_loss: 0.4452 - val_accuracy: 0.9091
Epoch 99/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4512 - accuracy: 0.9063 - val_loss: 0.4237 - val_accuracy: 0.9170
Epoch 100/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4493 - accuracy: 0.9072 - val_loss: 0.4305 - val_accuracy: 0.9176
> accuaracy of test set 90.619
```

In [ ]:

```
summarize_outcome(history = history)
```



In [ ]:

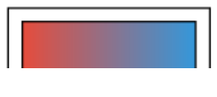
```
model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_11 (Conv2D)          (None, 32, 32, 32)        896

 max_pooling2d_7 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 conv2d_12 (Conv2D)          (None, 16, 16, 32)        9248

 max_pooling2d_8 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 flatten_6 (Flatten)         (None, 2048)              0

 dense_11 (Dense)            (None, 128)               262272

 dense_12 (Dense)            (None, 10)                1290

=================================================================
Total params: 273,706
Trainable params: 273,706
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
visualizer(model,format='png',view=True, filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model_L2norm')
Image(filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model_L2norm.png')
```

Out[ ]:



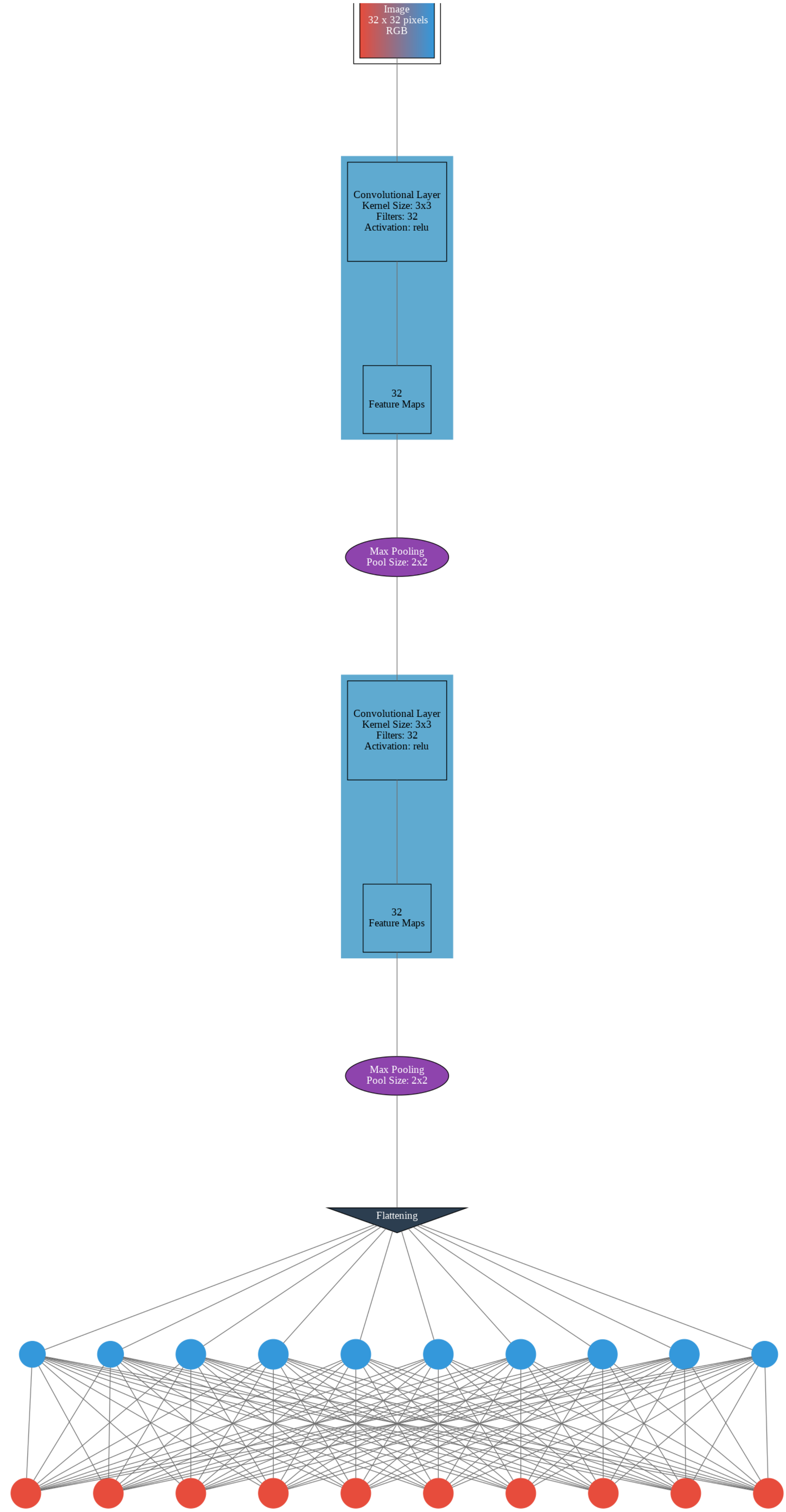


Image
32 x 32 pixels
RGB
Convolutional Layer
Kernel Size: 3x3
Filters: 32
Activation: relu
32
Feature Maps
Max Pooling
Pool Size: 2x2
Convolutional Layer
Kernel Size: 3x3
Filters: 32
Activation: relu
32
Feature Maps
Max Pooling
Pool Size: 2x2
Flattening
Units: 128 (+118 more)
Activation: relu
Output Units: 10
Activation: relu

Final Model with Dropout

In [ ]:

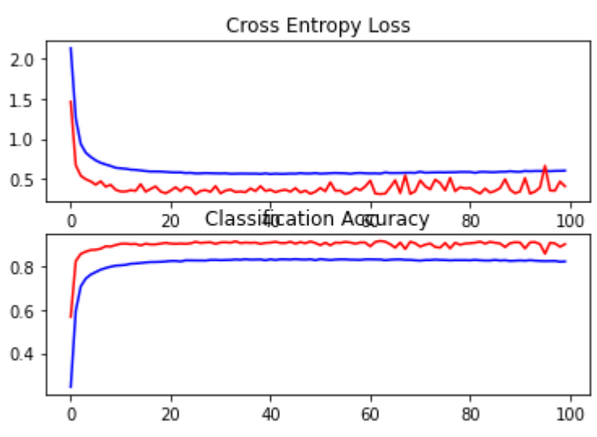
```
#Final Model with Dropout

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same',input_shape=(32, 32, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dropout(0.4))
model.add(Dense(10, activation='softmax'))
# compile model
model.compile(loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 54ms/step - loss: 2.1222 - accuracy: 0.2460 - val_loss: 1.4595 - val_accuracy: 0.5675
Epoch 2/100
487/487 [==============================] - 26s 54ms/step - loss: 1.2549 - accuracy: 0.5944 - val_loss: 0.6728 - val_accuracy: 0.8224
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.9360 - accuracy: 0.7065 - val_loss: 0.5426 - val_accuracy: 0.8554
Epoch 4/100
487/487 [==============================] - 26s 53ms/step - loss: 0.8285 - accuracy: 0.7421 - val_loss: 0.4989 - val_accuracy: 0.8658
Epoch 5/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7752 - accuracy: 0.7607 - val_loss: 0.4717 - val_accuracy: 0.8730
Epoch 6/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7324 - accuracy: 0.7720 - val_loss: 0.4308 - val_accuracy: 0.8744
Epoch 7/100
487/487 [==============================] - 26s 53ms/step - loss: 0.7019 - accuracy: 0.7833 - val_loss: 0.4736 - val_accuracy: 0.8792
Epoch 8/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6815 - accuracy: 0.7906 - val_loss: 0.4064 - val_accuracy: 0.8903
Epoch 9/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6628 - accuracy: 0.7969 - val_loss: 0.4296 - val_accuracy: 0.8890
Epoch 10/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6406 - accuracy: 0.8016 - val_loss: 0.3675 - val_accuracy: 0.8961
Epoch 11/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6339 - accuracy: 0.8030 - val_loss: 0.3467 - val_accuracy: 0.9017
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6284 - accuracy: 0.8051 - val_loss: 0.3465 - val_accuracy: 0.9025
Epoch 13/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6178 - accuracy: 0.8102 - val_loss: 0.3628 - val_accuracy: 0.8999
Epoch 14/100
487/487 [==============================] - 26s 53ms/step - loss: 0.6149 - accuracy: 0.8115 - val_loss: 0.3537 - val_accuracy: 0.9012
Epoch 15/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6077 - accuracy: 0.8136 - val_loss: 0.4376 - val_accuracy: 0.8943
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6002 - accuracy: 0.8162 - val_loss: 0.3441 - val_accuracy: 0.9022
Epoch 17/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5944 - accuracy: 0.8179 - val_loss: 0.3784 - val_accuracy: 0.8987
Epoch 18/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5947 - accuracy: 0.8183 - val_loss: 0.4098 - val_accuracy: 0.8990
Epoch 19/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5935 - accuracy: 0.8192 - val_loss: 0.3463 - val_accuracy: 0.9026
Epoch 20/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5898 - accuracy: 0.8207 - val_loss: 0.3245 - val_accuracy: 0.9062
Epoch 21/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5848 - accuracy: 0.8231 - val_loss: 0.3572 - val_accuracy: 0.9027
Epoch 22/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5828 - accuracy: 0.8232 - val_loss: 0.4004 - val_accuracy: 0.9015
Epoch 23/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5838 - accuracy: 0.8208 - val_loss: 0.3506 - val_accuracy: 0.9028
Epoch 24/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5757 - accuracy: 0.8260 - val_loss: 0.4025 - val_accuracy: 0.9015
Epoch 25/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5781 - accuracy: 0.8256 - val_loss: 0.3868 - val_accuracy: 0.9038
Epoch 26/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5705 - accuracy: 0.8253 - val_loss: 0.3128 - val_accuracy: 0.9107
Epoch 27/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5715 - accuracy: 0.8252 - val_loss: 0.3543 - val_accuracy: 0.9058
Epoch 28/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5751 - accuracy: 0.8248 - val_loss: 0.3640 - val_accuracy: 0.9075
Epoch 29/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5729 - accuracy: 0.8278 - val_loss: 0.3421 - val_accuracy: 0.9087
Epoch 30/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5715 - accuracy: 0.8278 - val_loss: 0.4137 - val_accuracy: 0.9015
Epoch 31/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5702 - accuracy: 0.8265 - val_loss: 0.3235 - val_accuracy: 0.9077
Epoch 32/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5723 - accuracy: 0.8274 - val_loss: 0.3579 - val_accuracy: 0.9087
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5656 - accuracy: 0.8277 - val_loss: 0.3704 - val_accuracy: 0.9066
Epoch 34/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5669 - accuracy: 0.8300 - val_loss: 0.3404 - val_accuracy: 0.9128
Epoch 35/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5684 - accuracy: 0.8285 - val_loss: 0.3462 - val_accuracy: 0.9051
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5684 - accuracy: 0.8313 - val_loss: 0.3374 - val_accuracy: 0.9074
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5667 - accuracy: 0.8290 - val_loss: 0.3851 - val_accuracy: 0.9061
Epoch 38/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5677 - accuracy: 0.8295 - val_loss: 0.3513 - val_accuracy: 0.9072
Epoch 39/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5639 - accuracy: 0.8300 - val_loss: 0.4128 - val_accuracy: 0.9015
Epoch 40/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5709 - accuracy: 0.8261 - val_loss: 0.3526 - val_accuracy: 0.9057
Epoch 41/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5622 - accuracy: 0.8307 - val_loss: 0.3683 - val_accuracy: 0.9072
Epoch 42/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5676 - accuracy: 0.8275 - val_loss: 0.3378 - val_accuracy: 0.9108
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5642 - accuracy: 0.8307 - val_loss: 0.3575 - val_accuracy: 0.9055
Epoch 44/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5650 - accuracy: 0.8306 - val_loss: 0.3677 - val_accuracy: 0.9053
Epoch 45/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5684 - accuracy: 0.8296 - val_loss: 0.3442 - val_accuracy: 0.9111
Epoch 46/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5673 - accuracy: 0.8316 - val_loss: 0.3890 - val_accuracy: 0.9024
Epoch 47/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5743 - accuracy: 0.8299 - val_loss: 0.3340 - val_accuracy: 0.9095
Epoch 48/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5695 - accuracy: 0.8293 - val_loss: 0.3585 - val_accuracy: 0.9044
Epoch 49/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5694 - accuracy: 0.8310 - val_loss: 0.3175 - val_accuracy: 0.9123
Epoch 50/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5750 - accuracy: 0.8264 - val_loss: 0.3441 - val_accuracy: 0.9054
Epoch 51/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5670 - accuracy: 0.8319 - val_loss: 0.3907 - val_accuracy: 0.8954
Epoch 52/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5701 - accuracy: 0.8294 - val_loss: 0.3491 - val_accuracy: 0.9071
Epoch 53/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5736 - accuracy: 0.8271 - val_loss: 0.4614 - val_accuracy: 0.8948
```

```
Epoch 54/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5761 - accuracy: 0.8295 - val_loss: 0.3556 - val_accuracy: 0.9073
Epoch 55/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5738 - accuracy: 0.8305 - val_loss: 0.3581 - val_accuracy: 0.9061
Epoch 56/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5726 - accuracy: 0.8304 - val_loss: 0.3140 - val_accuracy: 0.9105
Epoch 57/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5656 - accuracy: 0.8288 - val_loss: 0.3380 - val_accuracy: 0.9075
Epoch 58/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5743 - accuracy: 0.8285 - val_loss: 0.3889 - val_accuracy: 0.9021
Epoch 59/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5766 - accuracy: 0.8277 - val_loss: 0.3624 - val_accuracy: 0.9078
Epoch 60/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5731 - accuracy: 0.8304 - val_loss: 0.4094 - val_accuracy: 0.9075
Epoch 61/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5702 - accuracy: 0.8305 - val_loss: 0.4827 - val_accuracy: 0.8898
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5720 - accuracy: 0.8299 - val_loss: 0.3214 - val_accuracy: 0.9102
Epoch 63/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5682 - accuracy: 0.8297 - val_loss: 0.3151 - val_accuracy: 0.9149
Epoch 64/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5829 - accuracy: 0.8262 - val_loss: 0.3209 - val_accuracy: 0.9114
Epoch 65/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5755 - accuracy: 0.8289 - val_loss: 0.3996 - val_accuracy: 0.8998
Epoch 66/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5762 - accuracy: 0.8299 - val_loss: 0.4870 - val_accuracy: 0.8838
Epoch 67/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5767 - accuracy: 0.8303 - val_loss: 0.3260 - val_accuracy: 0.9079
Epoch 68/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5785 - accuracy: 0.8281 - val_loss: 0.5470 - val_accuracy: 0.8778
Epoch 69/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5796 - accuracy: 0.8280 - val_loss: 0.3165 - val_accuracy: 0.9111
Epoch 70/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5776 - accuracy: 0.8260 - val_loss: 0.3505 - val_accuracy: 0.9043
Epoch 71/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5917 - accuracy: 0.8247 - val_loss: 0.4854 - val_accuracy: 0.8887
Epoch 72/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5797 - accuracy: 0.8270 - val_loss: 0.4118 - val_accuracy: 0.8977
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5820 - accuracy: 0.8272 - val_loss: 0.3728 - val_accuracy: 0.9032
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5834 - accuracy: 0.8292 - val_loss: 0.4938 - val_accuracy: 0.8871
Epoch 75/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5842 - accuracy: 0.8276 - val_loss: 0.4508 - val_accuracy: 0.8840
Epoch 76/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5831 - accuracy: 0.8272 - val_loss: 0.3602 - val_accuracy: 0.9057
Epoch 77/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5852 - accuracy: 0.8257 - val_loss: 0.5154 - val_accuracy: 0.8804
Epoch 78/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5856 - accuracy: 0.8268 - val_loss: 0.3501 - val_accuracy: 0.9067
Epoch 79/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5890 - accuracy: 0.8266 - val_loss: 0.4001 - val_accuracy: 0.8973
Epoch 80/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5891 - accuracy: 0.8268 - val_loss: 0.3848 - val_accuracy: 0.9029
Epoch 81/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5884 - accuracy: 0.8260 - val_loss: 0.3902 - val_accuracy: 0.9045
Epoch 82/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5781 - accuracy: 0.8282 - val_loss: 0.3517 - val_accuracy: 0.9068
Epoch 83/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5897 - accuracy: 0.8260 - val_loss: 0.3211 - val_accuracy: 0.9117
Epoch 84/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5902 - accuracy: 0.8251 - val_loss: 0.3834 - val_accuracy: 0.9024
Epoch 85/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5864 - accuracy: 0.8251 - val_loss: 0.3326 - val_accuracy: 0.9106
Epoch 86/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5860 - accuracy: 0.8278 - val_loss: 0.3542 - val_accuracy: 0.9065
Epoch 87/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5913 - accuracy: 0.8250 - val_loss: 0.3887 - val_accuracy: 0.9034
Epoch 88/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5909 - accuracy: 0.8263 - val_loss: 0.4994 - val_accuracy: 0.8865
Epoch 89/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5979 - accuracy: 0.8233 - val_loss: 0.3698 - val_accuracy: 0.9044
Epoch 90/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5920 - accuracy: 0.8243 - val_loss: 0.3249 - val_accuracy: 0.9088
Epoch 91/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5907 - accuracy: 0.8255 - val_loss: 0.3461 - val_accuracy: 0.9077
Epoch 92/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6012 - accuracy: 0.8242 - val_loss: 0.5130 - val_accuracy: 0.8803
Epoch 93/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5975 - accuracy: 0.8265 - val_loss: 0.3200 - val_accuracy: 0.9082
Epoch 94/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6007 - accuracy: 0.8245 - val_loss: 0.3460 - val_accuracy: 0.9094
Epoch 95/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6003 - accuracy: 0.8232 - val_loss: 0.3957 - val_accuracy: 0.9002
Epoch 96/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6015 - accuracy: 0.8225 - val_loss: 0.6664 - val_accuracy: 0.8563
Epoch 97/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5984 - accuracy: 0.8229 - val_loss: 0.3595 - val_accuracy: 0.9067
Epoch 98/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6020 - accuracy: 0.8232 - val_loss: 0.3565 - val_accuracy: 0.9036
Epoch 99/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6027 - accuracy: 0.8196 - val_loss: 0.4707 - val_accuracy: 0.8879
Epoch 100/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6053 - accuracy: 0.8209 - val_loss: 0.4118 - val_accuracy: 0.9003
> accuaracy of test set 89.578
```

In [ ]:

```
summarize_outcome(history = history)
```



In [ ]:

```
model.summary()
```

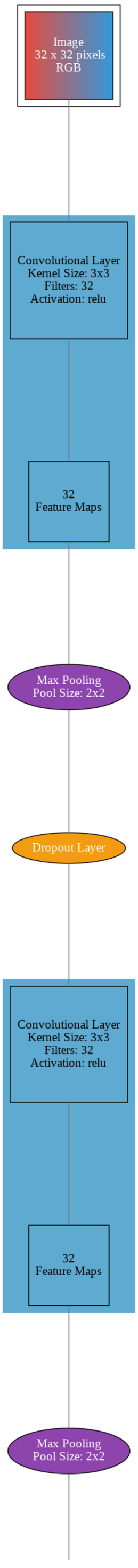
```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_13 (Conv2D)          (None, 32, 32, 32)        896

 max_pooling2d_9 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 dropout (Dropout)           (None, 16, 16, 32)        0

 conv2d_14 (Conv2D)          (None, 16, 16, 32)        9248

 max_pooling2d_10 (MaxPoolin  (None, 8, 8, 32)         0
```

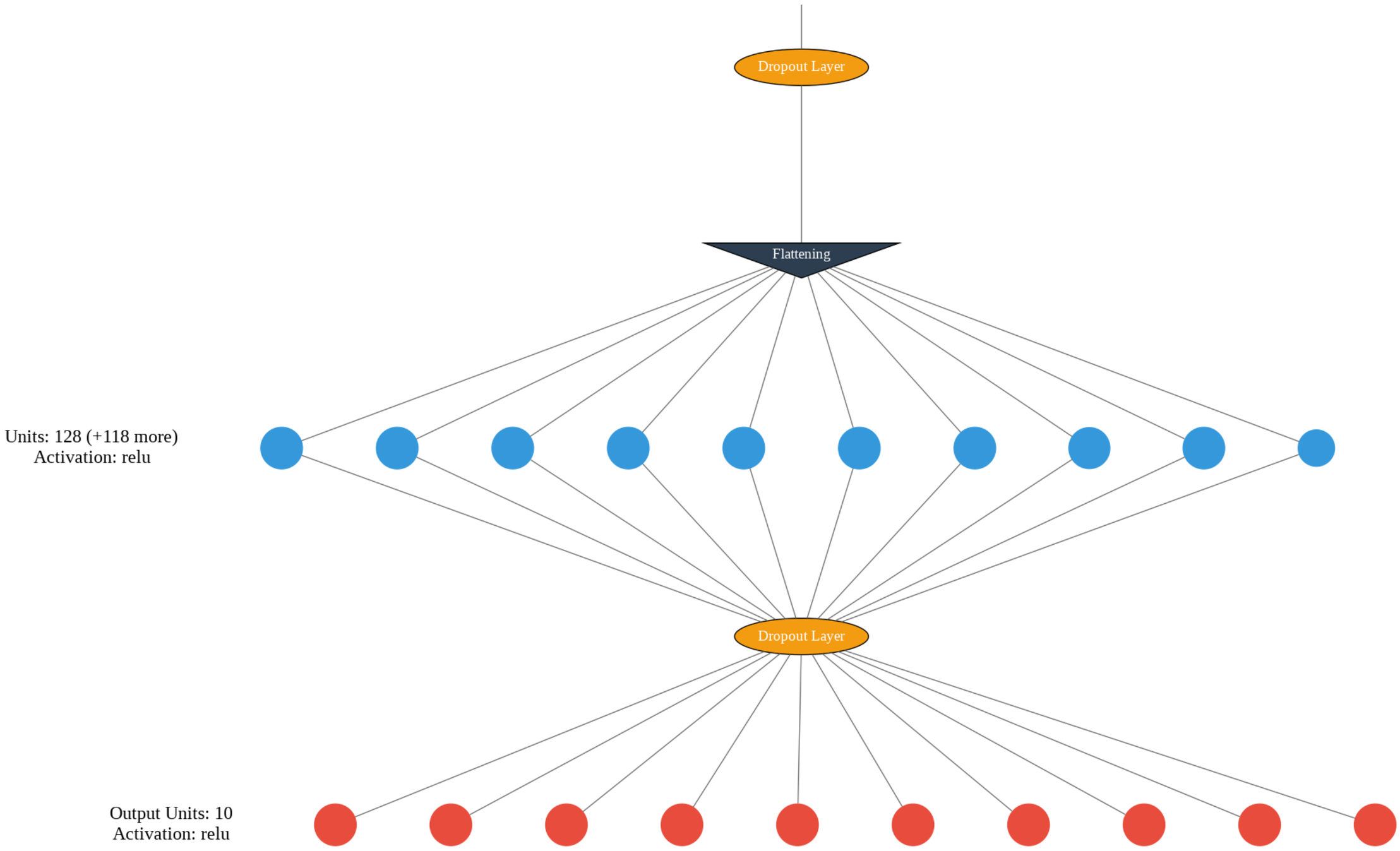
```
g2D)

dropout_1 (Dropout)          (None, 8, 8, 32)          0

flatten_7 (Flatten)          (None, 2048)              0

dense_13 (Dense)             (None, 128)               262272

dropout_2 (Dropout)          (None, 128)               0

dense_14 (Dense)             (None, 10)                1290

=================================================================
Total params: 273,706
Trainable params: 273,706
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
visualizer(model,format='png',view=True, filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model_Dropout')
Image(filename='/content/drive/MyDrive/Machine Learning/ML Bigdata/A3/model_Dropout.png')
```

Out[ ]:





Final model to find best learning rate

In [ ]:

```
#best model with optimizer finding best learning rate

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
#optimizer
lr_schedule = keras.callbacks.LearningRateScheduler(lambda epoch : 1e-4 * 10**(epoch/10))
optimizer = Adam(learning_rate= 1e-4, amsgrad=True)
# compile model
model.compile(optimizer=optimizer, loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val), callbacks = [lr_schedule])
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

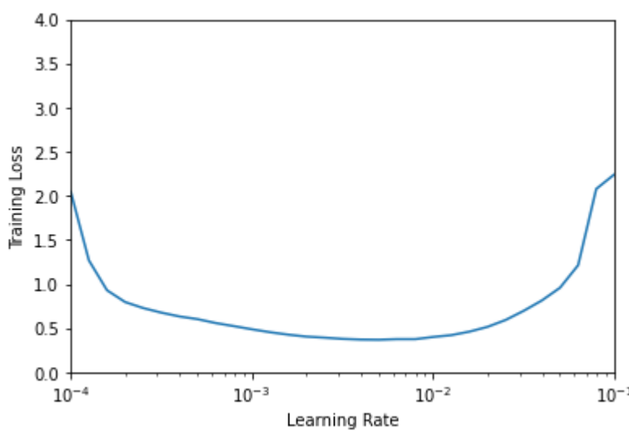
```
Epoch 1/100
487/487 [==============================] - 27s 54ms/step - loss: 2.0465 - accuracy: 0.2976 - val_loss: 1.5563 - val_accuracy: 0.5500 - lr: 1.0000e-04
Epoch 2/100
487/487 [==============================] - 26s 54ms/step - loss: 1.2697 - accuracy: 0.6258 - val_loss: 0.9142 - val_accuracy: 0.7655 - lr: 1.2589e-04
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.9311 - accuracy: 0.7285 - val_loss: 0.7092 - val_accuracy: 0.8070 - lr: 1.5849e-04
Epoch 4/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7976 - accuracy: 0.7668 - val_loss: 0.6141 - val_accuracy: 0.8352 - lr: 1.9953e-04
Epoch 5/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7295 - accuracy: 0.7887 - val_loss: 0.5684 - val_accuracy: 0.8432 - lr: 2.5119e-04
Epoch 6/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6777 - accuracy: 0.8043 - val_loss: 0.5133 - val_accuracy: 0.8622 - lr: 3.1623e-04
Epoch 7/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6344 - accuracy: 0.8188 - val_loss: 0.4860 - val_accuracy: 0.8706 - lr: 3.9811e-04
Epoch 8/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6039 - accuracy: 0.8279 - val_loss: 0.4613 - val_accuracy: 0.8733 - lr: 5.0119e-04
Epoch 9/100
487/487 [==============================] - 26s 53ms/step - loss: 0.5598 - accuracy: 0.8390 - val_loss: 0.4383 - val_accuracy: 0.8834 - lr: 6.3096e-04
Epoch 10/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5250 - accuracy: 0.8474 - val_loss: 0.4112 - val_accuracy: 0.8861 - lr: 7.9433e-04
Epoch 11/100
487/487 [==============================] - 27s 54ms/step - loss: 0.4891 - accuracy: 0.8576 - val_loss: 0.3910 - val_accuracy: 0.8921 - lr: 0.0010
Epoch 12/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4560 - accuracy: 0.8664 - val_loss: 0.3859 - val_accuracy: 0.8878 - lr: 0.0013
Epoch 13/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4284 - accuracy: 0.8738 - val_loss: 0.3549 - val_accuracy: 0.9005 - lr: 0.0016
Epoch 14/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4064 - accuracy: 0.8801 - val_loss: 0.3648 - val_accuracy: 0.8959 - lr: 0.0020
Epoch 15/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3945 - accuracy: 0.8821 - val_loss: 0.3623 - val_accuracy: 0.8953 - lr: 0.0025
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3808 - accuracy: 0.8853 - val_loss: 0.3423 - val_accuracy: 0.9012 - lr: 0.0032
Epoch 17/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3727 - accuracy: 0.8892 - val_loss: 0.3529 - val_accuracy: 0.8964 - lr: 0.0040
Epoch 18/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3704 - accuracy: 0.8881 - val_loss: 0.3676 - val_accuracy: 0.8948 - lr: 0.0050
Epoch 19/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3775 - accuracy: 0.8864 - val_loss: 0.3396 - val_accuracy: 0.9040 - lr: 0.0063
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3774 - accuracy: 0.8849 - val_loss: 0.3558 - val_accuracy: 0.8995 - lr: 0.0079
Epoch 21/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4023 - accuracy: 0.8783 - val_loss: 0.3517 - val_accuracy: 0.8966 - lr: 0.0100
Epoch 22/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4229 - accuracy: 0.8708 - val_loss: 0.4030 - val_accuracy: 0.8812 - lr: 0.0126
Epoch 23/100
487/487 [==============================] - 27s 55ms/step - loss: 0.4626 - accuracy: 0.8577 - val_loss: 0.4192 - val_accuracy: 0.8753 - lr: 0.0158
Epoch 24/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5168 - accuracy: 0.8439 - val_loss: 0.4116 - val_accuracy: 0.8836 - lr: 0.0200
Epoch 25/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5948 - accuracy: 0.8205 - val_loss: 0.5610 - val_accuracy: 0.8316 - lr: 0.0251
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6988 - accuracy: 0.7912 - val_loss: 0.5851 - val_accuracy: 0.8331 - lr: 0.0316
Epoch 27/100
487/487 [==============================] - 26s 54ms/step - loss: 0.8158 - accuracy: 0.7574 - val_loss: 0.7404 - val_accuracy: 0.8075 - lr: 0.0398
Epoch 28/100
```

```
487/487 [==============================] - 26s 54ms/step - loss: 0.9599 - accuracy: 0.7203 - val_loss: 0.7741 - val_accuracy: 0.7632 - lr: 0.0501
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 1.2150 - accuracy: 0.6416 - val_loss: 1.2472 - val_accuracy: 0.6396 - lr: 0.0631
Epoch 30/100
487/487 [==============================] - 26s 54ms/step - loss: 2.0791 - accuracy: 0.2661 - val_loss: 2.2369 - val_accuracy: 0.1949 - lr: 0.0794
Epoch 31/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2437 - accuracy: 0.1863 - val_loss: 2.2364 - val_accuracy: 0.1949 - lr: 0.1000
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2458 - accuracy: 0.1851 - val_loss: 2.2398 - val_accuracy: 0.1954 - lr: 0.1259
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2475 - accuracy: 0.1847 - val_loss: 2.2463 - val_accuracy: 0.1949 - lr: 0.1585
Epoch 34/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2494 - accuracy: 0.1796 - val_loss: 2.2475 - val_accuracy: 0.1949 - lr: 0.1995
Epoch 35/100
487/487 [==============================] - 27s 55ms/step - loss: 2.2510 - accuracy: 0.1803 - val_loss: 2.2497 - val_accuracy: 0.1431 - lr: 0.2512
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2541 - accuracy: 0.1818 - val_loss: 2.2403 - val_accuracy: 0.1949 - lr: 0.3162
Epoch 37/100
487/487 [==============================] - 27s 55ms/step - loss: 2.2570 - accuracy: 0.1778 - val_loss: 2.2866 - val_accuracy: 0.1431 - lr: 0.3981
Epoch 38/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2639 - accuracy: 0.1758 - val_loss: 2.2515 - val_accuracy: 0.1949 - lr: 0.5012
Epoch 39/100
487/487 [==============================] - 26s 54ms/step - loss: 765.4814 - accuracy: 0.1664 - val_loss: 2.2917 - val_accuracy: 0.1130 - lr: 0.6310
Epoch 40/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2718 - accuracy: 0.1733 - val_loss: 2.2536 - val_accuracy: 0.1949 - lr: 0.7943
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2796 - accuracy: 0.1705 - val_loss: 2.2566 - val_accuracy: 0.1431 - lr: 1.0000
Epoch 42/100
487/487 [==============================] - 26s 54ms/step - loss: 2.2897 - accuracy: 0.1656 - val_loss: 2.2484 - val_accuracy: 0.1949 - lr: 1.2589
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 2.3079 - accuracy: 0.1607 - val_loss: 2.2535 - val_accuracy: 0.1949 - lr: 1.5849
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 2.3288 - accuracy: 0.1564 - val_loss: 2.3107 - val_accuracy: 0.1949 - lr: 1.9953
Epoch 45/100
487/487 [==============================] - 27s 54ms/step - loss: 2.3328 - accuracy: 0.1534 - val_loss: 2.3374 - val_accuracy: 0.1431 - lr: 2.5119
Epoch 46/100
487/487 [==============================] - 27s 55ms/step - loss: 2.3845 - accuracy: 0.1454 - val_loss: 2.2850 - val_accuracy: 0.1431 - lr: 3.1623
Epoch 47/100
487/487 [==============================] - 27s 54ms/step - loss: 2.3917 - accuracy: 0.1461 - val_loss: 2.3977 - val_accuracy: 0.1949 - lr: 3.9811
Epoch 48/100
487/487 [==============================] - 27s 55ms/step - loss: 2.4276 - accuracy: 0.1412 - val_loss: 2.5897 - val_accuracy: 0.1130 - lr: 5.0119
Epoch 49/100
487/487 [==============================] - 26s 54ms/step - loss: 2.5055 - accuracy: 0.1362 - val_loss: 2.4800 - val_accuracy: 0.1043 - lr: 6.3096
Epoch 50/100
487/487 [==============================] - 27s 54ms/step - loss: 2.6022 - accuracy: 0.1332 - val_loss: 2.3752 - val_accuracy: 0.1949 - lr: 7.9433
Epoch 51/100
487/487 [==============================] - 27s 55ms/step - loss: 3.9661 - accuracy: 0.1274 - val_loss: 2.5030 - val_accuracy: 0.0753 - lr: 10.0000
Epoch 52/100
487/487 [==============================] - 26s 54ms/step - loss: 4.9185 - accuracy: 0.1204 - val_loss: 2.3896 - val_accuracy: 0.1949 - lr: 12.5893
Epoch 53/100
487/487 [==============================] - 26s 54ms/step - loss: 5.7158 - accuracy: 0.1171 - val_loss: 3.1862 - val_accuracy: 0.1949 - lr: 15.8489
Epoch 54/100
487/487 [==============================] - 27s 55ms/step - loss: 8.5309 - accuracy: 0.1131 - val_loss: 6.6127 - val_accuracy: 0.1043 - lr: 19.9526
Epoch 55/100
487/487 [==============================] - 26s 54ms/step - loss: 11.7979 - accuracy: 0.1123 - val_loss: 16.4063 - val_accuracy: 0.0753 - lr: 25.1189
Epoch 56/100
487/487 [==============================] - 26s 54ms/step - loss: 14.7494 - accuracy: 0.1123 - val_loss: 10.3444 - val_accuracy: 0.1949 - lr: 31.6228
Epoch 57/100
487/487 [==============================] - 27s 55ms/step - loss: 18.8415 - accuracy: 0.1150 - val_loss: 13.1866 - val_accuracy: 0.1431 - lr: 39.8107
Epoch 58/100
487/487 [==============================] - 27s 55ms/step - loss: 23.7395 - accuracy: 0.1150 - val_loss: 28.3780 - val_accuracy: 0.0968 - lr: 50.1187
Epoch 59/100
487/487 [==============================] - 26s 54ms/step - loss: 32.1511 - accuracy: 0.1154 - val_loss: 19.4790 - val_accuracy: 0.1130 - lr: 63.0957
Epoch 60/100
487/487 [==============================] - 27s 55ms/step - loss: 41.8136 - accuracy: 0.1137 - val_loss: 44.1479 - val_accuracy: 0.0968 - lr: 79.4328
Epoch 61/100
487/487 [==============================] - 26s 54ms/step - loss: 52.1699 - accuracy: 0.1142 - val_loss: 60.6354 - val_accuracy: 0.0734 - lr: 100.0000
Epoch 62/100
487/487 [==============================] - 26s 54ms/step - loss: 71.7755 - accuracy: 0.1161 - val_loss: 74.8685 - val_accuracy: 0.1431 - lr: 125.8925
Epoch 63/100
487/487 [==============================] - 27s 55ms/step - loss: 85.5486 - accuracy: 0.1146 - val_loss: 104.5940 - val_accuracy: 0.1431 - lr: 158.4893
Epoch 64/100
487/487 [==============================] - 26s 54ms/step - loss: 109.1521 - accuracy: 0.1158 - val_loss: 190.5076 - val_accuracy: 0.1043 - lr: 199.5262
Epoch 65/100
487/487 [==============================] - 26s 54ms/step - loss: 130.2039 - accuracy: 0.1140 - val_loss: 167.2367 - val_accuracy: 0.0968 - lr: 251.1886
Epoch 66/100
487/487 [==============================] - 27s 55ms/step - loss: 186.2167 - accuracy: 0.1167 - val_loss: 129.2554 - val_accuracy: 0.1949 - lr: 316.2278
Epoch 67/100
487/487 [==============================] - 27s 54ms/step - loss: 231.0633 - accuracy: 0.1124 - val_loss: 259.0487 - val_accuracy: 0.1431 - lr: 398.1072
Epoch 68/100
487/487 [==============================] - 27s 55ms/step - loss: 286.9827 - accuracy: 0.1157 - val_loss: 281.6281 - val_accuracy: 0.0734 - lr: 501.1872
Epoch 69/100
487/487 [==============================] - 27s 55ms/step - loss: 378.4511 - accuracy: 0.1151 - val_loss: 428.1575 - val_accuracy: 0.1043 - lr: 630.9573
Epoch 70/100
487/487 [==============================] - 27s 55ms/step - loss: 443.5919 - accuracy: 0.1143 - val_loss: 556.9639 - val_accuracy: 0.1949 - lr: 794.3282
Epoch 71/100
487/487 [==============================] - 27s 55ms/step - loss: 578.8078 - accuracy: 0.1130 - val_loss: 408.1056 - val_accuracy: 0.1431 - lr: 1000.0000
Epoch 72/100
487/487 [==============================] - 27s 55ms/step - loss: 704.4957 - accuracy: 0.1160 - val_loss: 408.4393 - val_accuracy: 0.1130 - lr: 1258.9254
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 868.2695 - accuracy: 0.1159 - val_loss: 924.7876 - val_accuracy: 0.0968 - lr: 1584.8932
Epoch 74/100
487/487 [==============================] - 26s 54ms/step - loss: 1164.0087 - accuracy: 0.1127 - val_loss: 1338.6044 - val_accuracy: 0.1949 - lr: 1995.2623
Epoch 75/100
487/487 [==============================] - 27s 55ms/step - loss: 1487.4097 - accuracy: 0.1160 - val_loss: 1135.1926 - val_accuracy: 0.0753 - lr: 2511.8865
Epoch 76/100
487/487 [==============================] - 26s 54ms/step - loss: 1801.3700 - accuracy: 0.1150 - val_loss: 1819.1163 - val_accuracy: 0.1431 - lr: 3162.2776
Epoch 77/100
487/487 [==============================] - 27s 55ms/step - loss: 2287.8730 - accuracy: 0.1149 - val_loss: 1879.0072 - val_accuracy: 0.1431 - lr: 3981.0718
Epoch 78/100
487/487 [==============================] - 27s 55ms/step - loss: 3087.9988 - accuracy: 0.1143 - val_loss: 4098.0254 - val_accuracy: 0.0753 - lr: 5011.8726
Epoch 79/100
487/487 [==============================] - 27s 55ms/step - loss: 3476.4258 - accuracy: 0.1124 - val_loss: 2546.5107 - val_accuracy: 0.1130 - lr: 6309.5732
Epoch 80/100
487/487 [==============================] - 27s 55ms/step - loss: 4479.0459 - accuracy: 0.1123 - val_loss: 3308.1401 - val_accuracy: 0.1431 - lr: 7943.2822
Epoch 81/100
487/487 [==============================] - 27s 55ms/step - loss: 6133.0308 - accuracy: 0.1149 - val_loss: 8536.3389 - val_accuracy: 0.1431 - lr: 10000.0000
Epoch 82/100
487/487 [==============================] - 27s 56ms/step - loss: 7686.8076 - accuracy: 0.1147 - val_loss: 9055.2568 - val_accuracy: 0.0734 - lr: 12589.2539
Epoch 83/100
487/487 [==============================] - 27s 55ms/step - loss: 9397.0195 - accuracy: 0.1134 - val_loss: 6590.4448 - val_accuracy: 0.1130 - lr: 15848.9316
Epoch 84/100
487/487 [==============================] - 27s 56ms/step - loss: 11289.2549 - accuracy: 0.1145 - val_loss: 12347.9492 - val_accuracy: 0.1949 - lr: 19952.6230
Epoch 85/100
487/487 [==============================] - 27s 55ms/step - loss: 14931.5381 - accuracy: 0.1160 - val_loss: 8480.6172 - val_accuracy: 0.0713 - lr: 25118.8652
Epoch 86/100
487/487 [==============================] - 27s 56ms/step - loss: 18725.8047 - accuracy: 0.1143 - val_loss: 21837.4785 - val_accuracy: 0.0753 - lr: 31622.7773
Epoch 87/100
487/487 [==============================] - 27s 55ms/step - loss: 24583.3105 - accuracy: 0.1151 - val_loss: 21809.8848 - val_accuracy: 0.1431 - lr: 39810.7188
Epoch 88/100
487/487 [==============================] - 27s 54ms/step - loss: 27911.4414 - accuracy: 0.1158 - val_loss: 14395.5332 - val_accuracy: 0.1043 - lr: 50118.7227
Epoch 89/100
487/487 [==============================] - 27s 55ms/step - loss: 34958.6992 - accuracy: 0.1141 - val_loss: 37964.0469 - val_accuracy: 0.0753 - lr: 63095.7344
Epoch 90/100
487/487 [==============================] - 27s 55ms/step - loss: 42220.9062 - accuracy: 0.1133 - val_loss: 73832.3984 - val_accuracy: 0.0968 - lr: 79432.8203
Epoch 91/100
487/487 [==============================] - 27s 55ms/step - loss: 60552.7734 - accuracy: 0.1129 - val_loss: 67340.9688 - val_accuracy: 0.1431 - lr: 100000.0000
Epoch 92/100
487/487 [==============================] - 27s 55ms/step - loss: 71903.5938 - accuracy: 0.1134 - val_loss: 65741.5781 - val_accuracy: 0.0753 - lr: 125892.5391
Epoch 93/100
487/487 [==============================] - 27s 55ms/step - loss: 98060.2188 - accuracy: 0.1140 - val_loss: 93241.7578 - val_accuracy: 0.0656 - lr: 158489.3125
Epoch 94/100
487/487 [==============================] - 27s 55ms/step - loss: 118027.2812 - accuracy: 0.1134 - val_loss: 126430.1719 - val_accuracy: 0.0968 - lr: 199526.2344
```

```
Epoch 95/100
487/487 [==============================] - 27s 55ms/step - loss: 147207.8906 - accuracy: 0.1170 - val_loss: 147710.6719 - val_accuracy: 0.1431 - lr: 251188.6406
Epoch 96/100
487/487 [==============================] - 27s 55ms/step - loss: 181546.5156 - accuracy: 0.1155 - val_loss: 67111.0625 - val_accuracy: 0.1130 - lr: 316227.7812
Epoch 97/100
487/487 [==============================] - 27s 55ms/step - loss: 228225.7969 - accuracy: 0.1132 - val_loss: 227701.6719 - val_accuracy: 0.0968 - lr: 398107.1562
Epoch 98/100
487/487 [==============================] - 27s 55ms/step - loss: 273445.8750 - accuracy: 0.1139 - val_loss: 408764.9688 - val_accuracy: 0.0713 - lr: 501187.2188
Epoch 99/100
487/487 [==============================] - 27s 55ms/step - loss: 357195.2188 - accuracy: 0.1139 - val_loss: 302213.9062 - val_accuracy: 0.1431 - lr: 630957.3750
Epoch 100/100
487/487 [==============================] - 27s 54ms/step - loss: 493985.7812 - accuracy: 0.1144 - val_loss: 562944.5000 - val_accuracy: 0.1949 - lr: 794328.2500
> accuaracy of test set 19.587
```

In [ ]:

```
#plotting Learning Rate vs Loss

plot.semilogx(history.history['lr'],history.history['loss'])
plot.axis([1e-4,1e-1,0,4])
plot.xlabel('Learning Rate')
plot.ylabel('Training Loss')
plot.show()
```



Final model with learning rate 1e-3

In [ ]:

```
#best model with learning rate 1e-3

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(32, 32, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
#optimizer
optimizer = Adam(learning_rate= 1e-3, amsgrad=True) #keras.optimizer.adam(lr_schedule = 1e-4, asmgrad = True)
# compile model
model.compile(optimizer=optimizer, loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 55ms/step - loss: 1.2869 - accuracy: 0.5857 - val_loss: 0.6517 - val_accuracy: 0.8227
Epoch 2/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7320 - accuracy: 0.7883 - val_loss: 0.5497 - val_accuracy: 0.8454
Epoch 3/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6335 - accuracy: 0.8163 - val_loss: 0.4701 - val_accuracy: 0.8732
Epoch 4/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5648 - accuracy: 0.8348 - val_loss: 0.4341 - val_accuracy: 0.8759
Epoch 5/100
487/487 [==============================] - 26s 54ms/step - loss: 0.5170 - accuracy: 0.8476 - val_loss: 0.4049 - val_accuracy: 0.8864
Epoch 6/100
487/487 [==============================] - 26s 53ms/step - loss: 0.4771 - accuracy: 0.8597 - val_loss: 0.3996 - val_accuracy: 0.8873
Epoch 7/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4477 - accuracy: 0.8668 - val_loss: 0.3758 - val_accuracy: 0.8956
Epoch 8/100
487/487 [==============================] - 26s 54ms/step - loss: 0.4208 - accuracy: 0.8751 - val_loss: 0.3665 - val_accuracy: 0.8968
Epoch 9/100
487/487 [==============================] - 26s 53ms/step - loss: 0.3970 - accuracy: 0.8817 - val_loss: 0.3582 - val_accuracy: 0.8981
Epoch 10/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3803 - accuracy: 0.8870 - val_loss: 0.3565 - val_accuracy: 0.8983
Epoch 11/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3654 - accuracy: 0.8909 - val_loss: 0.3468 - val_accuracy: 0.8994
Epoch 12/100
487/487 [==============================] - 26s 53ms/step - loss: 0.3550 - accuracy: 0.8945 - val_loss: 0.3462 - val_accuracy: 0.8997
Epoch 13/100
487/487 [==============================] - 26s 53ms/step - loss: 0.3403 - accuracy: 0.8985 - val_loss: 0.3282 - val_accuracy: 0.9075
Epoch 14/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3319 - accuracy: 0.9004 - val_loss: 0.3336 - val_accuracy: 0.9057
Epoch 15/100
487/487 [==============================] - 26s 53ms/step - loss: 0.3213 - accuracy: 0.9036 - val_loss: 0.3195 - val_accuracy: 0.9126
Epoch 16/100
487/487 [==============================] - 26s 54ms/step - loss: 0.3144 - accuracy: 0.9055 - val_loss: 0.3225 - val_accuracy: 0.9101
Epoch 17/100
487/487 [==============================] - 27s 55ms/step - loss: 0.3038 - accuracy: 0.9078 - val_loss: 0.3211 - val_accuracy: 0.9107
Epoch 18/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2996 - accuracy: 0.9087 - val_loss: 0.3181 - val_accuracy: 0.9112
Epoch 19/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2931 - accuracy: 0.9105 - val_loss: 0.3154 - val_accuracy: 0.9114
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2894 - accuracy: 0.9118 - val_loss: 0.3166 - val_accuracy: 0.9095
Epoch 21/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2830 - accuracy: 0.9150 - val_loss: 0.3184 - val_accuracy: 0.9116
Epoch 22/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2760 - accuracy: 0.9165 - val_loss: 0.3295 - val_accuracy: 0.9085
Epoch 23/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2727 - accuracy: 0.9177 - val_loss: 0.3244 - val_accuracy: 0.9112
Epoch 24/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2686 - accuracy: 0.9192 - val_loss: 0.3162 - val_accuracy: 0.9140
Epoch 25/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2637 - accuracy: 0.9211 - val_loss: 0.3069 - val_accuracy: 0.9144
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2596 - accuracy: 0.9208 - val_loss: 0.3093 - val_accuracy: 0.9147
Epoch 27/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2539 - accuracy: 0.9231 - val_loss: 0.3161 - val_accuracy: 0.9134
Epoch 28/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2463 - accuracy: 0.9249 - val_loss: 0.3183 - val_accuracy: 0.9134
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2448 - accuracy: 0.9270 - val_loss: 0.3141 - val_accuracy: 0.9171
Epoch 30/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2418 - accuracy: 0.9279 - val_loss: 0.3176 - val_accuracy: 0.9110
Epoch 31/100
487/487 [==============================] - 26s 53ms/step - loss: 0.2408 - accuracy: 0.9253 - val_loss: 0.3087 - val_accuracy: 0.9160
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2331 - accuracy: 0.9299 - val_loss: 0.3092 - val_accuracy: 0.9144
Epoch 33/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2277 - accuracy: 0.9305 - val_loss: 0.3332 - val_accuracy: 0.9095
Epoch 34/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2284 - accuracy: 0.9295 - val_loss: 0.3231 - val_accuracy: 0.9117
Epoch 35/100
```

```
487/487 [==============================] - 26s 54ms/step - loss: 0.2262 - accuracy: 0.9307 - val_loss: 0.3245 - val_accuracy: 0.9132
Epoch 36/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2230 - accuracy: 0.9327 - val_loss: 0.3262 - val_accuracy: 0.9094
Epoch 37/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2165 - accuracy: 0.9341 - val_loss: 0.3237 - val_accuracy: 0.9117
Epoch 38/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2143 - accuracy: 0.9348 - val_loss: 0.3192 - val_accuracy: 0.9141
Epoch 39/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2144 - accuracy: 0.9352 - val_loss: 0.3300 - val_accuracy: 0.9110
Epoch 40/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2061 - accuracy: 0.9367 - val_loss: 0.3239 - val_accuracy: 0.9133
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2038 - accuracy: 0.9384 - val_loss: 0.3463 - val_accuracy: 0.9121
Epoch 42/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2032 - accuracy: 0.9378 - val_loss: 0.3231 - val_accuracy: 0.9131
Epoch 43/100
487/487 [==============================] - 26s 54ms/step - loss: 0.2030 - accuracy: 0.9389 - val_loss: 0.3308 - val_accuracy: 0.9115
Epoch 44/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1990 - accuracy: 0.9390 - val_loss: 0.3311 - val_accuracy: 0.9166
Epoch 45/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1955 - accuracy: 0.9393 - val_loss: 0.3201 - val_accuracy: 0.9134
Epoch 46/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1962 - accuracy: 0.9389 - val_loss: 0.3267 - val_accuracy: 0.9114
Epoch 47/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1902 - accuracy: 0.9416 - val_loss: 0.3347 - val_accuracy: 0.9156
Epoch 48/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1923 - accuracy: 0.9405 - val_loss: 0.3459 - val_accuracy: 0.9078
Epoch 49/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1862 - accuracy: 0.9423 - val_loss: 0.3250 - val_accuracy: 0.9113
Epoch 50/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1858 - accuracy: 0.9427 - val_loss: 0.3360 - val_accuracy: 0.9138
Epoch 51/100
487/487 [==============================] - 27s 55ms/step - loss: 0.1878 - accuracy: 0.9416 - val_loss: 0.3381 - val_accuracy: 0.9139
Epoch 52/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1808 - accuracy: 0.9435 - val_loss: 0.3384 - val_accuracy: 0.9122
Epoch 53/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1788 - accuracy: 0.9446 - val_loss: 0.3416 - val_accuracy: 0.9135
Epoch 54/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1786 - accuracy: 0.9441 - val_loss: 0.3348 - val_accuracy: 0.9166
Epoch 55/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1730 - accuracy: 0.9448 - val_loss: 0.3427 - val_accuracy: 0.9141
Epoch 56/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1739 - accuracy: 0.9458 - val_loss: 0.3562 - val_accuracy: 0.9075
Epoch 57/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1720 - accuracy: 0.9462 - val_loss: 0.3411 - val_accuracy: 0.9144
Epoch 58/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1684 - accuracy: 0.9472 - val_loss: 0.3423 - val_accuracy: 0.9148
Epoch 59/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1696 - accuracy: 0.9466 - val_loss: 0.3406 - val_accuracy: 0.9138
Epoch 60/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1662 - accuracy: 0.9477 - val_loss: 0.3544 - val_accuracy: 0.9104
Epoch 61/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1653 - accuracy: 0.9484 - val_loss: 0.3388 - val_accuracy: 0.9137
Epoch 62/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1637 - accuracy: 0.9480 - val_loss: 0.3459 - val_accuracy: 0.9136
Epoch 63/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1631 - accuracy: 0.9480 - val_loss: 0.3354 - val_accuracy: 0.9141
Epoch 64/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1611 - accuracy: 0.9495 - val_loss: 0.3473 - val_accuracy: 0.9099
Epoch 65/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1606 - accuracy: 0.9504 - val_loss: 0.3545 - val_accuracy: 0.9119
Epoch 66/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1564 - accuracy: 0.9508 - val_loss: 0.3501 - val_accuracy: 0.9142
Epoch 67/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1571 - accuracy: 0.9502 - val_loss: 0.3477 - val_accuracy: 0.9086
Epoch 68/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1556 - accuracy: 0.9500 - val_loss: 0.3526 - val_accuracy: 0.9124
Epoch 69/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1560 - accuracy: 0.9509 - val_loss: 0.3593 - val_accuracy: 0.9163
Epoch 70/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1512 - accuracy: 0.9523 - val_loss: 0.3627 - val_accuracy: 0.9109
Epoch 71/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1526 - accuracy: 0.9518 - val_loss: 0.3602 - val_accuracy: 0.9151
Epoch 72/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1526 - accuracy: 0.9520 - val_loss: 0.3632 - val_accuracy: 0.9126
Epoch 73/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1503 - accuracy: 0.9523 - val_loss: 0.3557 - val_accuracy: 0.9122
Epoch 74/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1469 - accuracy: 0.9537 - val_loss: 0.3591 - val_accuracy: 0.9144
Epoch 75/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1424 - accuracy: 0.9548 - val_loss: 0.3623 - val_accuracy: 0.9146
Epoch 76/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1454 - accuracy: 0.9539 - val_loss: 0.3518 - val_accuracy: 0.9117
Epoch 77/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1479 - accuracy: 0.9532 - val_loss: 0.3629 - val_accuracy: 0.9096
Epoch 78/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1421 - accuracy: 0.9557 - val_loss: 0.3600 - val_accuracy: 0.9125
Epoch 79/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1406 - accuracy: 0.9564 - val_loss: 0.3709 - val_accuracy: 0.9094
Epoch 80/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1416 - accuracy: 0.9544 - val_loss: 0.3716 - val_accuracy: 0.9096
Epoch 81/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1403 - accuracy: 0.9556 - val_loss: 0.3808 - val_accuracy: 0.9111
Epoch 82/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1375 - accuracy: 0.9565 - val_loss: 0.3856 - val_accuracy: 0.9077
Epoch 83/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1387 - accuracy: 0.9565 - val_loss: 0.3747 - val_accuracy: 0.9115
Epoch 84/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1370 - accuracy: 0.9567 - val_loss: 0.3923 - val_accuracy: 0.9078
Epoch 85/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1358 - accuracy: 0.9576 - val_loss: 0.3960 - val_accuracy: 0.9114
Epoch 86/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1353 - accuracy: 0.9568 - val_loss: 0.3820 - val_accuracy: 0.9087
Epoch 87/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1333 - accuracy: 0.9584 - val_loss: 0.3920 - val_accuracy: 0.9094
Epoch 88/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1354 - accuracy: 0.9570 - val_loss: 0.3837 - val_accuracy: 0.9105
Epoch 89/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1321 - accuracy: 0.9580 - val_loss: 0.3962 - val_accuracy: 0.9054
Epoch 90/100
487/487 [==============================] - 26s 53ms/step - loss: 0.1313 - accuracy: 0.9588 - val_loss: 0.3940 - val_accuracy: 0.9074
Epoch 91/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1316 - accuracy: 0.9591 - val_loss: 0.3794 - val_accuracy: 0.9122
Epoch 92/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1281 - accuracy: 0.9598 - val_loss: 0.4035 - val_accuracy: 0.9087
Epoch 93/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1314 - accuracy: 0.9573 - val_loss: 0.3871 - val_accuracy: 0.9119
Epoch 94/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1264 - accuracy: 0.9600 - val_loss: 0.3908 - val_accuracy: 0.9077
Epoch 95/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1298 - accuracy: 0.9585 - val_loss: 0.3931 - val_accuracy: 0.9108
Epoch 96/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1279 - accuracy: 0.9596 - val_loss: 0.4003 - val_accuracy: 0.9107
Epoch 97/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1279 - accuracy: 0.9590 - val_loss: 0.3923 - val_accuracy: 0.9114
Epoch 98/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1226 - accuracy: 0.9606 - val_loss: 0.3938 - val_accuracy: 0.9065
Epoch 99/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1201 - accuracy: 0.9611 - val_loss: 0.4134 - val_accuracy: 0.9110
Epoch 100/100
487/487 [==============================] - 26s 54ms/step - loss: 0.1201 - accuracy: 0.9615 - val_loss: 0.4024 - val_accuracy: 0.9096
> accuaracy of test set 89.006
```

Final Model with all hyper tunings

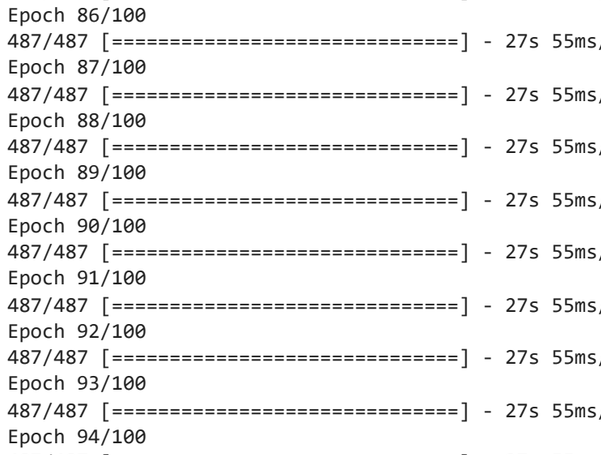
In [ ]:
```
#Final Model

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same',kernel_regularizer = l2(0.001), input_shape=(32, 32, 3)))
model.add(BatchNormalization())
model.add(AveragePooling2D((2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same',kernel_regularizer = l2(0.001)))
model.add(BatchNormalization())
model.add(AveragePooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform',kernel_regularizer = l2(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.4))
model.add(Dense(10, activation='softmax'))
#optimizer
optimizer = Adam(learning_rate= 1e-3, amsgrad=True) #keras.optimizer.adam(lr_schedule = 1e-4, asmgrad = True)
# compile model
model.compile(optimizer=optimizer, loss='categorical_crossentropy',   metrics=['accuracy'])
# fit model
history = model.fit(train_datagen.flow(X_train, y_train, batch_size=128),epochs=100, validation_data=(X_val, y_val))
# evaluate model
_, acc = model.evaluate(test_imgs, test_labels, verbose=0)
print('> accuaracy of test set %.3f' % (acc * 100.0))
```

```
Epoch 1/100
487/487 [==============================] - 27s 55ms/step - loss: 1.9073 - accuracy: 0.4985 - val_loss: 1.0688 - val_accuracy: 0.8000
Epoch 2/100
487/487 [==============================] - 26s 54ms/step - loss: 1.1033 - accuracy: 0.7446 - val_loss: 0.7565 - val_accuracy: 0.8656
Epoch 3/100
487/487 [==============================] - 27s 55ms/step - loss: 0.9336 - accuracy: 0.7888 - val_loss: 0.6290 - val_accuracy: 0.8881
Epoch 4/100
487/487 [==============================] - 27s 55ms/step - loss: 0.8511 - accuracy: 0.8074 - val_loss: 0.6358 - val_accuracy: 0.8825
Epoch 5/100
487/487 [==============================] - 26s 54ms/step - loss: 0.8112 - accuracy: 0.8175 - val_loss: 0.5843 - val_accuracy: 0.8990
Epoch 6/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7795 - accuracy: 0.8237 - val_loss: 0.5646 - val_accuracy: 0.8986
Epoch 7/100
487/487 [==============================] - 26s 54ms/step - loss: 0.7612 - accuracy: 0.8307 - val_loss: 0.5395 - val_accuracy: 0.9102
Epoch 8/100
487/487 [==============================] - 27s 54ms/step - loss: 0.7503 - accuracy: 0.8344 - val_loss: 0.5470 - val_accuracy: 0.9032
Epoch 9/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7345 - accuracy: 0.8392 - val_loss: 0.5367 - val_accuracy: 0.9080
Epoch 10/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7328 - accuracy: 0.8382 - val_loss: 0.5331 - val_accuracy: 0.9082
Epoch 11/100
487/487 [==============================] - 27s 54ms/step - loss: 0.7222 - accuracy: 0.8407 - val_loss: 0.5282 - val_accuracy: 0.9061
Epoch 12/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7140 - accuracy: 0.8432 - val_loss: 0.6169 - val_accuracy: 0.8824
Epoch 13/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7139 - accuracy: 0.8425 - val_loss: 0.5802 - val_accuracy: 0.9014
Epoch 14/100
487/487 [==============================] - 27s 54ms/step - loss: 0.7051 - accuracy: 0.8456 - val_loss: 0.5407 - val_accuracy: 0.8974
Epoch 15/100
487/487 [==============================] - 27s 54ms/step - loss: 0.6978 - accuracy: 0.8464 - val_loss: 0.5187 - val_accuracy: 0.9097
Epoch 16/100
487/487 [==============================] - 27s 55ms/step - loss: 0.7019 - accuracy: 0.8459 - val_loss: 0.5077 - val_accuracy: 0.9136
Epoch 17/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6920 - accuracy: 0.8472 - val_loss: 0.5162 - val_accuracy: 0.9095
Epoch 18/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6942 - accuracy: 0.8473 - val_loss: 0.5176 - val_accuracy: 0.9098
Epoch 19/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6912 - accuracy: 0.8489 - val_loss: 0.4969 - val_accuracy: 0.9179
Epoch 20/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6852 - accuracy: 0.8506 - val_loss: 0.4952 - val_accuracy: 0.9142
Epoch 21/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6841 - accuracy: 0.8511 - val_loss: 0.5095 - val_accuracy: 0.9100
Epoch 22/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6853 - accuracy: 0.8505 - val_loss: 0.5546 - val_accuracy: 0.8949
Epoch 23/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6825 - accuracy: 0.8501 - val_loss: 0.5068 - val_accuracy: 0.9119
Epoch 24/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6825 - accuracy: 0.8494 - val_loss: 0.4828 - val_accuracy: 0.9190
Epoch 25/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6756 - accuracy: 0.8529 - val_loss: 0.4915 - val_accuracy: 0.9159
Epoch 26/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6703 - accuracy: 0.8544 - val_loss: 0.4820 - val_accuracy: 0.9186
Epoch 27/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6725 - accuracy: 0.8536 - val_loss: 0.4855 - val_accuracy: 0.9153
Epoch 28/100
487/487 [==============================] - 27s 54ms/step - loss: 0.6735 - accuracy: 0.8539 - val_loss: 0.4877 - val_accuracy: 0.9156
Epoch 29/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6666 - accuracy: 0.8558 - val_loss: 0.4856 - val_accuracy: 0.9204
Epoch 30/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6687 - accuracy: 0.8535 - val_loss: 0.4793 - val_accuracy: 0.9168
Epoch 31/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6720 - accuracy: 0.8520 - val_loss: 0.4882 - val_accuracy: 0.9159
Epoch 32/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6677 - accuracy: 0.8550 - val_loss: 0.4820 - val_accuracy: 0.9152
Epoch 33/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6657 - accuracy: 0.8536 - val_loss: 0.5032 - val_accuracy: 0.9106
Epoch 34/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6652 - accuracy: 0.8542 - val_loss: 0.4718 - val_accuracy: 0.9202
Epoch 35/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6622 - accuracy: 0.8558 - val_loss: 0.4841 - val_accuracy: 0.9133
Epoch 36/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6633 - accuracy: 0.8550 - val_loss: 0.5513 - val_accuracy: 0.8911
Epoch 37/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6591 - accuracy: 0.8553 - val_loss: 0.4685 - val_accuracy: 0.9219
Epoch 38/100
487/487 [==============================] - 27s 54ms/step - loss: 0.6591 - accuracy: 0.8560 - val_loss: 0.4692 - val_accuracy: 0.9206
Epoch 39/100
487/487 [==============================] - 27s 54ms/step - loss: 0.6584 - accuracy: 0.8567 - val_loss: 0.4873 - val_accuracy: 0.9146
Epoch 40/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6532 - accuracy: 0.8578 - val_loss: 0.4713 - val_accuracy: 0.9184
Epoch 41/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6540 - accuracy: 0.8579 - val_loss: 0.4778 - val_accuracy: 0.9157
Epoch 42/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6566 - accuracy: 0.8580 - val_loss: 0.4823 - val_accuracy: 0.9167
Epoch 43/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6596 - accuracy: 0.8559 - val_loss: 0.5001 - val_accuracy: 0.9133
Epoch 44/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6522 - accuracy: 0.8576 - val_loss: 0.5022 - val_accuracy: 0.9143
Epoch 45/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6550 - accuracy: 0.8576 - val_loss: 0.4744 - val_accuracy: 0.9168
Epoch 46/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6498 - accuracy: 0.8581 - val_loss: 0.4735 - val_accuracy: 0.9197
Epoch 47/100
487/487 [==============================] - 26s 54ms/step - loss: 0.6558 - accuracy: 0.8577 - val_loss: 0.4880 - val_accuracy: 0.9145
Epoch 48/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6485 - accuracy: 0.8588 - val_loss: 0.4781 - val_accuracy: 0.9176
Epoch 49/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6511 - accuracy: 0.8593 - val_loss: 0.4597 - val_accuracy: 0.9219
Epoch 50/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6423 - accuracy: 0.8602 - val_loss: 0.4778 - val_accuracy: 0.9151
Epoch 51/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6423 - accuracy: 0.8600 - val_loss: 0.4725 - val_accuracy: 0.9206
Epoch 52/100
```

```
487/487 [==============================] - 27s 55ms/step - loss: 0.6502 - accuracy: 0.8584 - val_loss: 0.5070 - val_accuracy: 0.9123
Epoch 53/100
487/487 [==============================] - 27s 54ms/step - loss: 0.6473 - accuracy: 0.8591 - val_loss: 0.4916 - val_accuracy: 0.9093
Epoch 54/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6470 - accuracy: 0.8581 - val_loss: 0.4975 - val_accuracy: 0.9146
Epoch 55/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6442 - accuracy: 0.8615 - val_loss: 0.4823 - val_accuracy: 0.9162
Epoch 56/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6448 - accuracy: 0.8584 - val_loss: 0.4804 - val_accuracy: 0.9168
Epoch 57/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6490 - accuracy: 0.8574 - val_loss: 0.4878 - val_accuracy: 0.9168
Epoch 58/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6430 - accuracy: 0.8601 - val_loss: 0.4769 - val_accuracy: 0.9151
Epoch 59/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6408 - accuracy: 0.8598 - val_loss: 0.4710 - val_accuracy: 0.9216
Epoch 60/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6455 - accuracy: 0.8603 - val_loss: 0.5071 - val_accuracy: 0.9052
Epoch 61/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6483 - accuracy: 0.8569 - val_loss: 0.4826 - val_accuracy: 0.9168
Epoch 62/100
487/487 [==============================] - 27s 54ms/step - loss: 0.6445 - accuracy: 0.8606 - val_loss: 0.4645 - val_accuracy: 0.9215
Epoch 63/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6390 - accuracy: 0.8603 - val_loss: 0.4719 - val_accuracy: 0.9176
Epoch 64/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6411 - accuracy: 0.8603 - val_loss: 0.4626 - val_accuracy: 0.9216
Epoch 65/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6415 - accuracy: 0.8610 - val_loss: 0.4577 - val_accuracy: 0.9211
Epoch 66/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6364 - accuracy: 0.8622 - val_loss: 0.4840 - val_accuracy: 0.9152
Epoch 67/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6375 - accuracy: 0.8615 - val_loss: 0.4490 - val_accuracy: 0.9257
Epoch 68/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6424 - accuracy: 0.8597 - val_loss: 0.4589 - val_accuracy: 0.9218
Epoch 69/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6418 - accuracy: 0.8589 - val_loss: 0.4583 - val_accuracy: 0.9205
Epoch 70/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6386 - accuracy: 0.8607 - val_loss: 0.4763 - val_accuracy: 0.9161
Epoch 71/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6369 - accuracy: 0.8608 - val_loss: 0.4702 - val_accuracy: 0.9183
Epoch 72/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6326 - accuracy: 0.8611 - val_loss: 0.4719 - val_accuracy: 0.9137
Epoch 73/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6368 - accuracy: 0.8610 - val_loss: 0.5160 - val_accuracy: 0.9039
Epoch 74/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6367 - accuracy: 0.8618 - val_loss: 0.4780 - val_accuracy: 0.9115
Epoch 75/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6361 - accuracy: 0.8606 - val_loss: 0.4609 - val_accuracy: 0.9196
Epoch 76/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6355 - accuracy: 0.8613 - val_loss: 0.4755 - val_accuracy: 0.9178
Epoch 77/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6382 - accuracy: 0.8600 - val_loss: 0.4886 - val_accuracy: 0.9106
Epoch 78/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6321 - accuracy: 0.8630 - val_loss: 0.4612 - val_accuracy: 0.9201
Epoch 79/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6377 - accuracy: 0.8599 - val_loss: 0.4678 - val_accuracy: 0.9193
Epoch 80/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6375 - accuracy: 0.8619 - val_loss: 0.4747 - val_accuracy: 0.9177
Epoch 81/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6404 - accuracy: 0.8587 - val_loss: 0.4795 - val_accuracy: 0.9183
Epoch 82/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6319 - accuracy: 0.8633 - val_loss: 0.4808 - val_accuracy: 0.9152
Epoch 83/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6367 - accuracy: 0.8609 - val_loss: 0.4882 - val_accuracy: 0.9115
Epoch 84/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6295 - accuracy: 0.8628 - val_loss: 0.4680 - val_accuracy: 0.9181
Epoch 85/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6275 - accuracy: 0.8625 - val_loss: 0.4635 - val_accuracy: 0.9210
Epoch 86/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6312 - accuracy: 0.8621 - val_loss: 0.4864 - val_accuracy: 0.9157
Epoch 87/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6320 - accuracy: 0.8614 - val_loss: 0.4608 - val_accuracy: 0.9240
Epoch 88/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6274 - accuracy: 0.8634 - val_loss: 0.4699 - val_accuracy: 0.9143
Epoch 89/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6278 - accuracy: 0.8636 - val_loss: 0.4751 - val_accuracy: 0.9189
Epoch 90/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6323 - accuracy: 0.8625 - val_loss: 0.4602 - val_accuracy: 0.9213
Epoch 91/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6269 - accuracy: 0.8628 - val_loss: 0.4811 - val_accuracy: 0.9137
Epoch 92/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6310 - accuracy: 0.8625 - val_loss: 0.4633 - val_accuracy: 0.9165
Epoch 93/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6286 - accuracy: 0.8636 - val_loss: 0.4747 - val_accuracy: 0.9155
Epoch 94/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6310 - accuracy: 0.8626 - val_loss: 0.4567 - val_accuracy: 0.9242
Epoch 95/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6309 - accuracy: 0.8630 - val_loss: 0.4899 - val_accuracy: 0.9136
Epoch 96/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6303 - accuracy: 0.8621 - val_loss: 0.4632 - val_accuracy: 0.9193
Epoch 97/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6316 - accuracy: 0.8608 - val_loss: 0.4572 - val_accuracy: 0.9222
Epoch 98/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6260 - accuracy: 0.8628 - val_loss: 0.5021 - val_accuracy: 0.9059
Epoch 99/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6235 - accuracy: 0.8647 - val_loss: 0.4548 - val_accuracy: 0.9222
Epoch 100/100
487/487 [==============================] - 27s 55ms/step - loss: 0.6286 - accuracy: 0.8631 - val_loss: 0.4867 - val_accuracy: 0.9131
> accuaracy of test set 91.261
```

In [ ]:

```
summarize_outcome(history = history)
```



In [ ]:

```
model.summary()
```

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_19 (Conv2D)          (None, 32, 32, 32)        896

 batch_normalization_3 (Batc  (None, 32, 32, 32)       128
 hNormalization)

 average_pooling2d_2 (Averag  (None, 16, 16, 32)       0
 ePooling2D)
```

```
 dropout_3 (Dropout)         (None, 16, 16, 32)        0

 conv2d_20 (Conv2D)          (None, 16, 16, 32)        9248

 batch_normalization_4 (Batc  (None, 16, 16, 32)       128
 hNormalization)

 average_pooling2d_3 (Averag  (None, 8, 8, 32)         0
 ePooling2D)

 dropout_4 (Dropout)         (None, 8, 8, 32)          0

 flatten_10 (Flatten)        (None, 2048)              0

 dense_19 (Dense)            (None, 128)               262272

 batch_normalization_5 (Batc  (None, 128)              512
 hNormalization)

 dropout_5 (Dropout)         (None, 128)               0

 dense_20 (Dense)            (None, 10)                1290

=================================================================
Total params: 274,474
Trainable params: 274,090
Non-trainable params: 384
_________________________________________________________________
```

In [ ]: