Name:Merna Adel Ragab
ID:4168
DR/Nour-ElDin Ismail

# Skin Cancer

Skin cancer is the most common human malignancy, is primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis, a biopsy and histopathological examination. Automated classification of skin lesions using images is a challenging task owing to the fine-grained variability in the appearance of skin lesion.

This the HAM10000 ("Human Against Machine with 10000 training images") dataset.It consists of 10015 dermatoscopicimages which are released as a training set for academic machine learning purposes

It has 7 different classes of skin cancer which are listed below :

1. **Melanocytic nevi**
2. **Melanoma**
3. **Benign keratosis-like lesions**
4. **Basal cell carcinoma**
5. **Actinic keratoses**
6. **Vascular lesions**
7. **Dermatofibroma**

I have followed following 14 steps for model building and evaluation which are as follows :

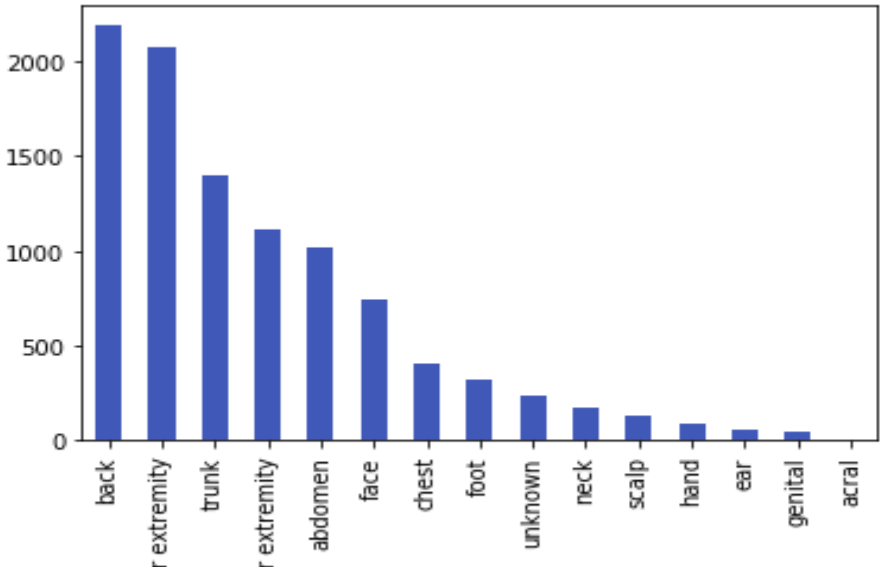
**Step 1 : Importing Essential Libraries**
**Step 2: Making Dictionary of images and labels**
**Step 3: Reading and Processing Data**
**Step 4: Data Cleaning**
**Step 5: Exploratory data analysis (EDA)**
**Step 6: Loading & Resizing of images**
**Step 7: Train Test Split**
**Step 8: Normalization**
**Step 9: Label Encoding**
**Step 10: Train validation split**
**Step 11: Model Building (CNN)**
**Step 12: Setting Optimizer & Annealing**
**Step 13: Fitting the model**
**Step 14: Model Evaluation**

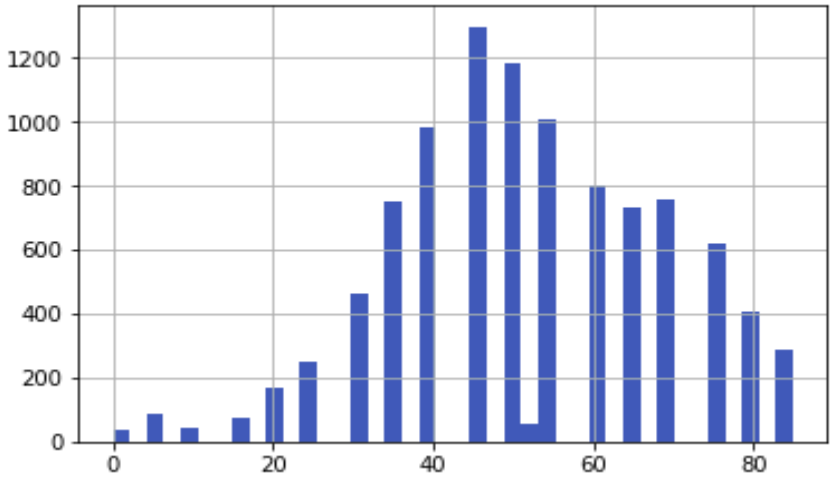
# Dataset:

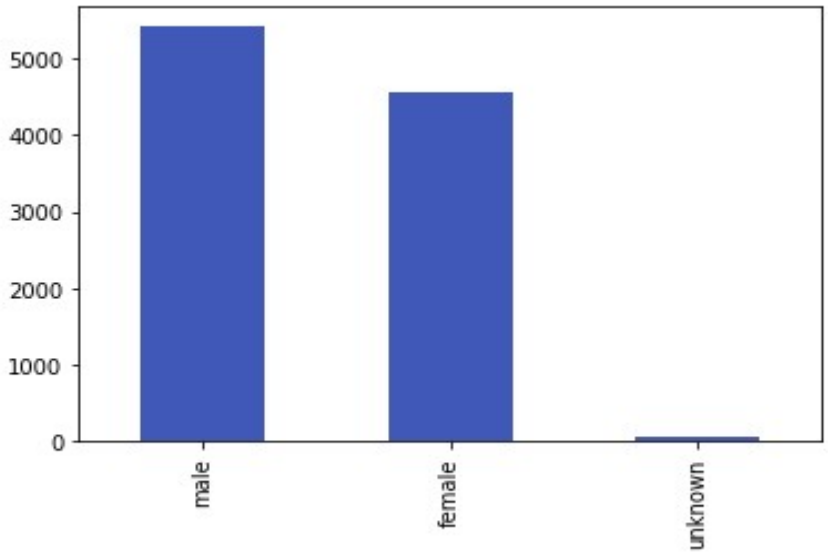
## Step 3 : Reading & Processing data

| | lesion_id | image_id | dx | dx_type | age | sex | localization | path | cell_type | cell_type_idx |
|---|---|---|---|---|---|---|---|---|---|---|
| **9725** | HAM_0004376 | ISIC_0024843 | akiec | histo | 70.0 | female | face | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Actinic keratoses | 0 |
| **6059** | HAM_0003024 | ISIC_0024768 | nv | follow_up | 35.0 | female | trunk | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **4540** | HAM_0001659 | ISIC_0026564 | nv | follow_up | 35.0 | male | lower extremity | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **3817** | HAM_0004625 | ISIC_0029346 | nv | follow_up | 40.0 | male | upper extremity | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **7914** | HAM_0000443 | ISIC_0034271 | nv | histo | 35.0 | female | back | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **8910** | HAM_0007176 | ISIC_0032144 | nv | histo | 30.0 | male | lower extremity | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **799** | HAM_0007355 | ISIC_0024705 | bkl | confocal | 45.0 | female | face | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Benign keratosis-like lesions | 2 |
| **4304** | HAM_0003147 | ISIC_0029168 | nv | follow_up | 50.0 | female | trunk | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **8203** | HAM_0007585 | ISIC_0032347 | nv | histo | 35.0 | female | back | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |
| **8822** | HAM_0004058 | ISIC_0030802 | nv | histo | 80.0 | female | lower extremity | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | 4 |

Histograms:


2000
1500
1000
500
0
back
r extremity
trunk
r extremity
abdomen
face
chest
foot
unknown
neck
scalp
hand
ear
genital
acral


1200
1000
800
600
400
200
0
0
20
40
60
80


5000
4000
3000
2000
1000
0
male
female
unknown

| | lesion_id | image_id | dx | dx_type | age | sex | localization | path | cell_type | image |
|---|---|---|---|---|---|---|---|---|---|---|
| **4007** | HAM_0001798 | ISIC_0030141 | nv | follow_up | 50.0 | male | back | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | [[[236, 159, 177], [239, 162, 180], [238, 164,... |
| **8796** | HAM_0004836 | ISIC_0026674 | nv | histo | 35.0 | male | lower extremity | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | [[[192, 181, 195], [192, 179, 196], [193, 180,... |
| **7692** | HAM_0006496 | ISIC_0032579 | nv | histo | 55.0 | female | lower extremity | ../input/skin-cancer-mnist-ham10000/HAM10000_i... | Melanocytic nevi | [[[169, 159, 193], [169, 160, 187], [166, 158,... |

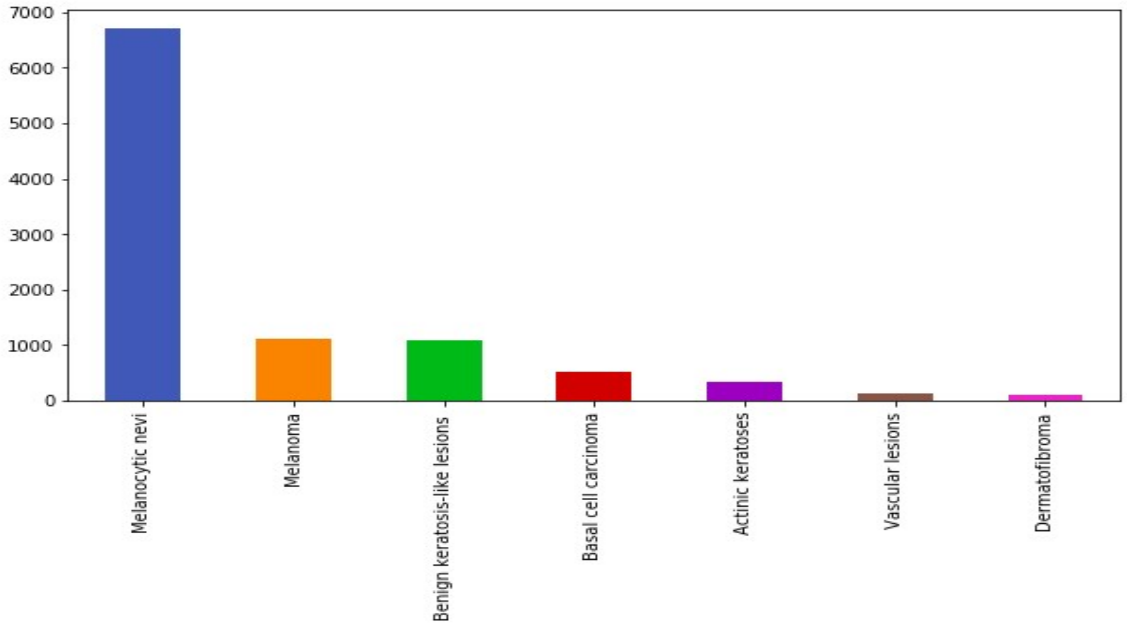
# Step 4 : Data Cleaning

In this step we check for Missing values and datatype of each field

# Step 5 : EDA

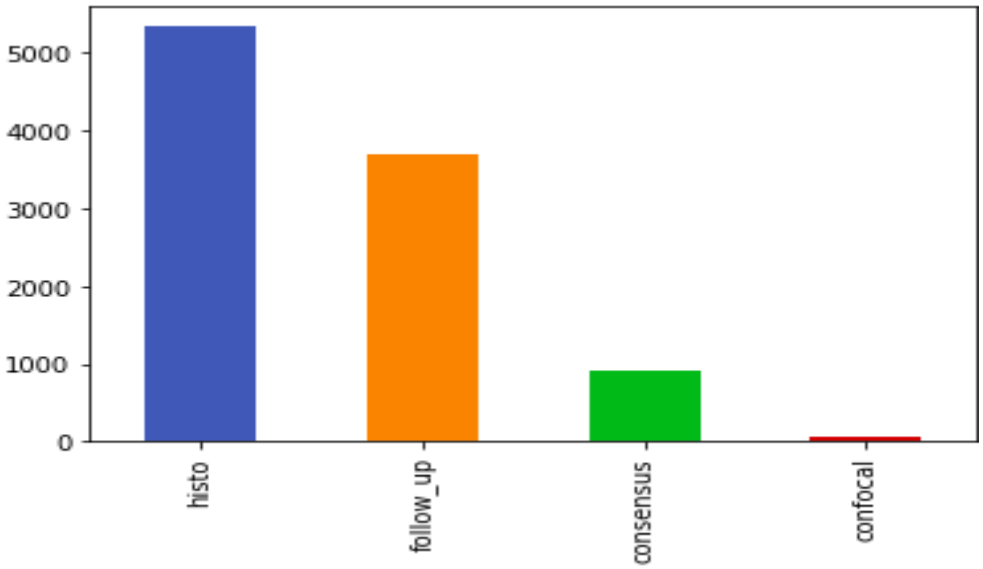
In this we will explore different features of the dataset , their distrubtions and actual counts

Plot to see distribution of 7 different classes of cell type




Its seems from the above plot that in this dataset cell type Melanecytic nevi has very large number of instances in comparison to other cell types

Plotting of Technical Validation field (ground truth) which is dx_type to see the distribution of its 4 categories which are listed below :
**1. Histopathology(Histo):** Histopathologic diagnoses of excised lesions have been performed by specialized dermatopathologists.
**2. Confocal:** Reflectance confocal microscopy is an in-vivo imaging technique with a resolution at near-cellular level , and some facial benign with a grey-world assumption of all training-set images in Lab-color space before and after manual histogram changes.
**3. Follow-up:** If nevi monitored by digital dermatoscopy did not show any changes during 3 follow-up visits or 1.5 years biologists accepted this as evidence of biologic benignity. Only nevi, but no other benign diagnoses were labeled with this type of ground-truth because dermatologists usually do not monitor dermatofibromas, seborrheic keratoses, or vascular lesions.
**4. Consensus:** For typical benign cases without histopathology or followup biologists provide an expert-consensus rating of authors PT and HK. They applied the consensus label only if both authors independently gave the same unequivocal benign diagnosis. Lesions with this type of groundtruth were usually photographed for educational reasons and did not need further follow-up or biopsy for confirmation.

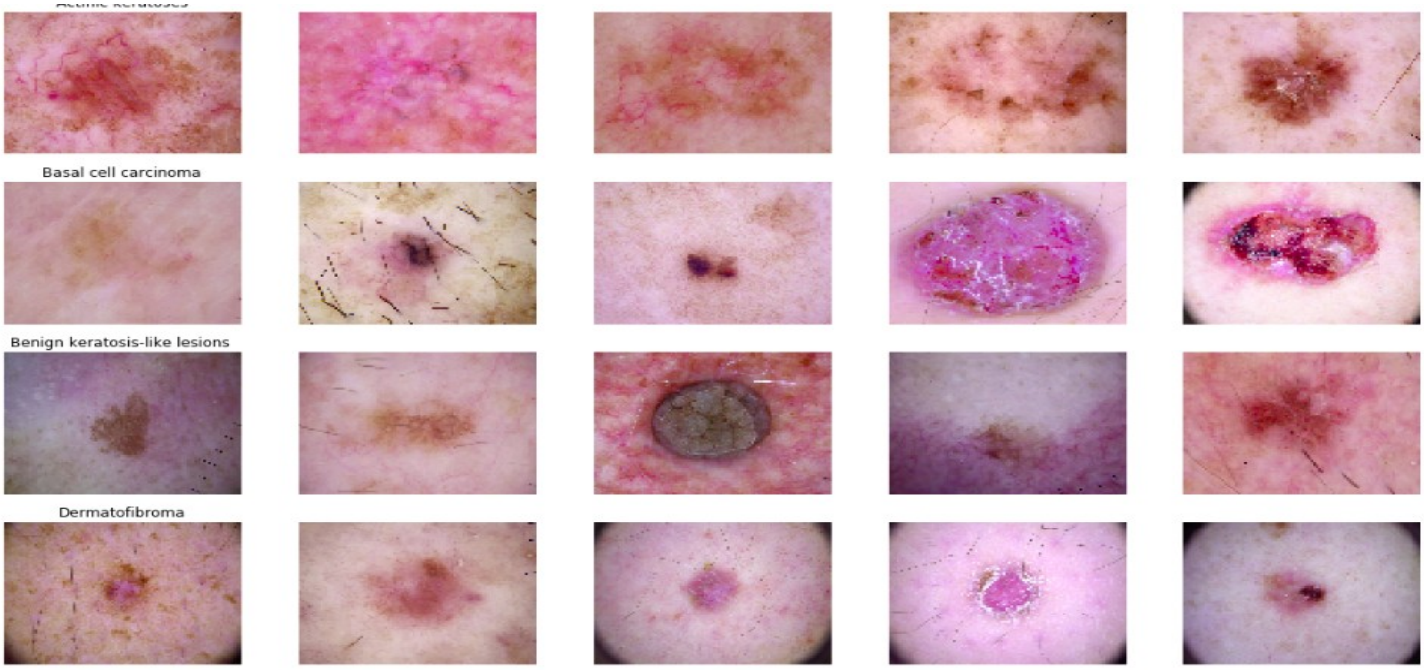



# Step 6: Loading and resizing of images

450 x 600 x3

to

(75, 100, 3)  10015




## Step 7 : Train Test Split

In this step we have splitted the dataset into training and testing set of 80:20 ratio

## Step 8 : Normalization

I choosed to normalize the x_train, x_test by substracting from theor mean values and then dividing by thier standard deviation.

## Step 9 : Label Encoding

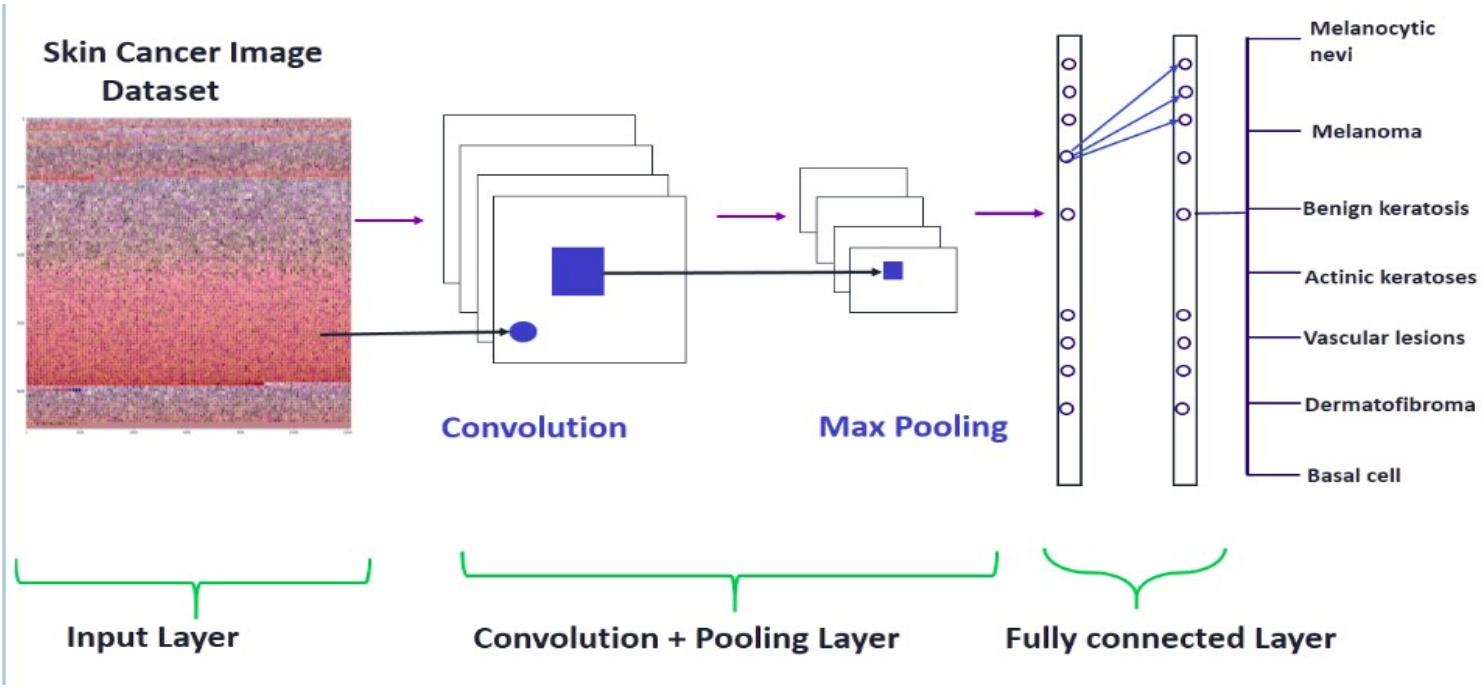
Labels are 7 different classes of skin cancer types from 0 to 6. We need to encode these lables to one hot vectors

## Step 10 : Splitting training and validation split

I choosed to split the train set in two parts : a small fraction (10%) became the validation set which the model is evaluated and the rest (90%) is used to train the model.

# Step 11: Model Building

## CNN




I used the Keras Sequential API, where you have just to add one layer at a time, starting from the input.
The first is the convolutional (Conv2D) layer. It is like a set of learnable filters. I choosed to set 32 filters for the two firsts conv2D layers and 64 filters for the two last ones. Each filter transforms a part of the image (defined by the kernel size) using the kernel filter. The kernel filter matrix is applied on the whole image. Filters can be seen as a transformation of the image.
The CNN can isolate features that are useful everywhere from these transformed images (feature maps).
The second important layer in CNN is the pooling (MaxPool2D) layer. This layer simply acts as a downsampling filter. It looks at the 2 neighboring pixels and picks the maximal value. These are used to reduce computational cost, and to some extent also reduce overfitting. We have to choose the pooling size (i.e the area size pooled each time) more the pooling dimension is high, more the downsampling is important.
Combining convolutional and pooling layers, CNN are able to combine local features and learn more global features of the image.

Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their wieghts to zero) for each training sample. This drops randomly a propotion of the network and forces the network to learn features in a distributed way. This technique also improves generalization and reduces the overfitting.
'relu' is the rectifier (activation function max(0,x). The rectifier activation function is used to add non linearity to the network.
The Flatten layer is use to convert the final feature maps into a one single 1D vector. This flattening step is needed so that you can make use of fully connected layers after some convolutional/maxpool layers. It combines all the found local features of the previous convolutional layers.
In the end i used the features in two fully-connected (Dense) layers which is just artificial an neural networks (ANN) classifier. In the last layer(Dense(10,activation="softmax")) the net outputs distribution of probability of each class.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 75, 100, 32)       896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 75, 100, 32)       9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 37, 50, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 37, 50, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 37, 50, 64)        18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 37, 50, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 18, 25, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 18, 25, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 28800)             0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               3686528
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 903
=================================================================
Total params: 3,752,999
Trainable params: 3,752,999
Non-trainable params: 0
```

# Step 12: Setting Optimizer and Annealer

Once our layers are added to the model, we need to set up a score function, a loss function and an optimisation algorithm.

# Step 13: Fitting the model

In this step finally I fit the model into x_train, y_train. In this step I have choosen batch size of 10 and 50 epochs as small as your batch size will be more efficiently your model will train and I have choosen 50 epochs to give the model sufficient epochs to train

```
Epoch 1/50
721/721 [==============================] - 23s 32ms/step - loss: 1.0150 - acc: 0.6666 - val_loss: 0.8938 - val_acc: 0.6845
Epoch 2/50
721/721 [==============================] - 15s 21ms/step - loss: 0.9164 - acc: 0.6714 - val_loss: 0.8716 - val_acc: 0.6820
Epoch 3/50
721/721 [==============================] - 15s 21ms/step - loss: 0.8927 - acc: 0.6716 - val_loss: 0.8333 - val_acc: 0.6958
Epoch 4/50
721/721 [==============================] - 17s 23ms/step - loss: 0.8618 - acc: 0.6850 - val_loss: 0.8176 - val_acc: 0.7120
Epoch 5/50
721/721 [==============================] - 15s 21ms/step - loss: 0.8271 - acc: 0.6925 - val_loss: 0.7529 - val_acc: 0.7257
Epoch 6/50
721/721 [==============================] - 15s 21ms/step - loss: 0.8286 - acc: 0.7010 - val_loss: 0.8308 - val_acc: 0.7195
Epoch 7/50
721/721 [==============================] - 16s 22ms/step - loss: 0.7884 - acc: 0.7136 - val_loss: 0.7454 - val_acc: 0.7344
Epoch 8/50
721/721 [==============================] - 16s 22ms/step - loss: 0.7781 - acc: 0.7165 - val_loss: 0.7104 - val_acc: 0.7394
Epoch 9/50
```

# Step 14: Model Evaluation

In this step we will check the testing accuracy and validation accuracy of our model,plot confusion matrix and also check the missclassified images count of each type

```
2003/2003 [==============================] - 1s 263us/step
802/802 [==============================] - 0s 230us/step
Validation: accuracy = 0.773067  ;  loss_v = 0.626629
Test: accuracy = 0.760359  ;  loss = 0.620183
```