This is a to-be submitted sample document

Social Media Image Classification using CNN

1. Abstract

In today's world, social media is redefining the way we discover people and new ideas, concepts and information. With the original idea of bringing friends and family close, social media platforms today are being extensively used for business purposes, spreading ideas and concepts through blogs and other publishing networks, finding career opportunities, bringing together people with the same thoughts and much more. Due to this, tons of images are uploaded on social media everyday and many of them make their way in our mobiles phones as a result of media share. This requires an urgent need to study and develop effective ways to segregate images from one another to more effectively manage the plethora of digital images on social media. Recent developments in the fields of Deep Learning and Image Processing have paved way to study and understand intricate details within an image In this paper, we aim to develop an image classifier with the help of Convolutional Neural Network by training it on a self modelled dataset to divide images into three broad categories- document based images, quote based images and photographs.
Keywords: Deep Learning, Convolutional Neural Network

2. Introduction

In recent times, there has been a burst of photo sharing via social media platforms, mainly Whatsapp. Being a part of Whatsapp groups compliments with a large number of photos landing in our memory, irrespective of whether we need those images in the near future or not. We aim to develop an image classification technique which would classify the images in three main categories, namely Quote based images, Photographs and Documents.Once the classification is done, various actions can be performed on these images including but not limited to deleting quote based images, organising document based images and managing redundant images. It can also help to manage and study the images that are shared over social media platforms in a more efficient and effective manner.

Convolutional Neural Network (CNN) has been extensively used for many pattern recognition problems, especially image recognition by many researchers including (Zha, Luisier, Andrews, Srivastava, & Salakhutdinov, 2015, p. 60.1),(Li et al., 2014, p. 845), (Tianyu, Zhenjiang, & Jianhu, 2018, p. 556) etc. In this paper, we propose a CNN based network to create a classification model. The basic network consists of a series of convolutional layers with Batch Normalisation(Ioffe, 2015) with Rectified Linear Unit (ReLU) as activation function. A dropout

(Srivastava, 2020) setting with rate 0.5 is used to fight over-fitting. The model also uses ResNet34 as a pre-trained model to utilize a portion of pre-trained weights. During the early stages of training, the parameters in the initial layers are not updated, In the later stages,all the layers are used for training. The network here is divided into layer groups with each layer having their own learning rate. The concept of superconvergence(Smith, L. N., & Topin, N,2019) is used to train the layers faster, An overall accuracy of 94.8% is achieved by the classifier.

3. Literature Review

(Krizhevsky, Sutskever, & Hinton, 2017a, p. 88) trained a deepConvolutional Neural Network (CNN) which acted as the biggest breakthrough in the field of CNN. Their deep CNN was used to classify 1.2 million high resolution images which were there in the ImageNet LSVRC-2010 contest into 1000 different classes. They contradicted the earlier accepted norm by the computer vision researchers that a neural network can not be successfully used to classify images by simply providing input of images and classifying them into labels by acquiring all the knowledge from the provided training data. Using the neural network which had 60 million parameters and 650,000 neurons, with 5 convolutional layers, some followed by max-pooling layers and 3 fully connected layers with a final 1000-way softmax and a regularisation method to reduce overfitting called dropout, test model achieved top-1 and top-5 error
rates of 37.5% and 17.0%, respectively. A variant of this model was used again in the ILSVRC-2012 competition and achieved a mindblowing error rate of 15.3% with the second best model achieving 26.2% error rate.

But one thing that was not noticed in the work by (Krizhevsky, Sutskever, & Hinton, 2017a, p. 88) was a clear understanding of why the classifier actually did such a good job or what can be done to improve. (Zeiler, 2013) addressed these issues by using a visualisation technique to get insights regarding the functions of the intermediate feature layers and the operation of the classifier. These could also be used to find model architecture better than the Krizhevsky et al. classification model.

Since then, CNN has been used for image classification in a variety of fields. (Li et al., 2014, p. 845) used CNN for Image Patch Classification, which is considered to be an important task in medical imaging applications. A customized CNN with a shallow convolutional layer was designed to classify lung images patches with interstitial lung disease (ILD). CNN automatically learned the intrinsic image features from lung patches to solve the classification problem. The same architecture was also claimed to be used to perform other medical image classification tasks.

(Zha, Luisier, Andrews, Srivastava, & Salakhutdinov, 2015, p. 60.1) used CNNs trained for image classification for event detection in videos. Using astute choice of dimensions led to a significant increase in performance in comparison to previous approaches that were used.

With the use of deep convolutional neural networks (D-CNNs) used by (Krizhevsky, Sutskever, & Hinton, 2017a, p. 88), researchers have created a lot of variants of D-CNNs. But, almost all of these were trained on the giant ImageNet datasets. Datasets like CIFAR-10 rarely took advantage of the depth of the models because of overfitting.(Liu & Deng, 2015, p. 733) proposed a modified VGC-16 network, in which by adding stronger regularizer and Batch Normalization, they achieved an error rate of 8.45% on CIFAR-10 without severe overfitting. This was an important publication for it brought forward the idea that if a model is strong enough to fit a large dataset, it can fit a small dataset too.

Due to these advancements and refinements that took place, CNNs became a favorite for solving complex image classification problems and opened up a whole new dimension of use. For example, (Levine, Pastor, Krizhevsky, Ibarz, & Quillen, 2017, p. 427) used CNN to create a hand eye coordination for robotic grasping. A large CNN was trained using only monocular images to predict the probability that a task space motion of a robotic gripper will lead to a successful grasp. The network learned the spatial relationship between the gripper and the objects, thus learning the hand eye coordination and used it to perform successful grisps in real life.
Similarly, (Kolsch, Afzal, Ebbecke, & Liwicki, 2017, p. 1321) presented a way for real time training and testing for document image classification. They used a deep network for feature extraction followed by Extreme Learning Machines (ELM) for classification. This modification was done to reduce the time consumed in the training process.

In the following years, a lot of modifications and extra functionalities were used with the CNN to perform image classification. (Tianyu, Zhenjiang, & Jianhu, 2018, p. 556) mentioned that even though CNN have achieved high performance in the field of image classification, hand crafted features are still very important as they help to describe images from specific aspects and help CNN in classification tasks. They explored features fusion methods and proposed methods to combine CNN with the hand crafted features. Their results showed that CNN performed better when the hand crafted features were combined.

Similarly, (Zhang et al., 2018, p. 143) integrated the two well-recognised neural network algorithm, the NN and the pixel-based Multilayer Perceptron (MLP) for classification of very fine spatial resolution (VFSR) remotely sensed imagery. The effectiveness of this combination was tested in both urban and rural areas using aerial photography and it consistently outperformed MLP and CNN in terms of classification accuracy. CNN was also used by (Chen

& Tao, 2018, p. 628) for Polarimetric synthetic aperture radar (PolSAR) image classification with limited training data and maintaining generalization performance.

(Song et al., 2018, p. 6026) brought forward the problem that CNN have performed wonderfully on single-label image classification but the use of CNN for multi-label images is still a problem. They proposed a deep multi-modal CNN for multi-instance multi-label image classification, called MMCNN-MIML. By putting together CNN and multi-instance multi-label (MIML) learning, the model is able to represent each image as a bunch of instances for image classification and has the merits of both CNNs and MIML.

Usually in the state-of-the-art CNNs, their architecture is manually built. Thus, it makes it difficult for people with limited knowledge to create their own architecture to suit their image classification problem. (Sun, 2018) proposed a way to automatically generate CNN architecture using genetic algorithms. This allows users to not have a strong domain knowledge of the CNN when using the proposed algorithm,while they can still get a good CNN architecture for their image classification problem.

After analysing the past work done in the field of image classification , CNN has emerged as the top priority for image classification for it has given successful results in recent times. We thus choose to use CNN for our image classification problem.

Author	**Title of the article**	**Focus Areas**	**Main Idea**
Krizhevsky et al., 2017a, p. 88	ImageNet classification with deep convolutional neural networks	Image Recognition, Convolutional Neural Network	Created one of the first image classifier using deep CNN, classifying over 1.2 million images into 1000 different classes with an top-1 and top-5 error rates of 37.5% and 17.0%, respectively
Zeiler, 2013	Visualizing and Understanding Convolutional Networks	Multi-layered Deconvolutional Network (deconvnet)	Created a visualisation technique to better understand the classifier's working and to improve the model.

Li et al., 2014, p. 845	Medical image classification with convolutional neural network	CNN in medical domain	Use the CNN architecture to classify lung images patches with interstitial lung disease (ILD).
Zha et al., 2015, p. 60.1	Exploiting Image-trained CNN Architectures for Unconstrained Video Classification	CNN	Used CNN based image classifier for event detection in videos.
Liu et al., 2015, p. 733	Very deep convolutional neural network based image classification using small training sample size	Deep CNNs	Using a small training data to fit a D-CNN without overfitting.
Levine et al. 2017, p. 427	Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection	CNN	Used CNN to teach a robotic gripper hand eye coordination for successful gripping in real life.
Kolsch et al. 2017, p. 1321	Real-Time Document Image Classification Using Deep CNN and Extreme Learning Machines	Deep CNN and ELM	Created a document image classification using Deep CNN and ELM.
Tianyu et al., 2018, p. 556	Combining CNN with Hand-Crafted Features for Image Classification	CNN and Hand Crafted Features	Created a framework to fuse hand crafted features with CNN to get better result

Zhang et al., 2018, p. 143	A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification	CNN and MLP	Combined MLP and CNN for classification of very fine spatial resolution (VFSR) remotely sensed imagery
Radenović, 2017	Fine-tuning CNN Image Retrieval with No Human Annotation	CNN	Proposed to fine tune CNN model for unordered images in a fully automated manner
Chen et al., 2018, p. 628	PolSAR Image Classification Using Polarimetric-Feature-Driven Deep Convolutional Neural Network	CNN	Used Polarimetric synthetic aperture radar (PolSAR) image classification with limited training data and maintaining generalization performance.
Song et al., 2018, p. 6026	A Deep Multi-Modal CNN for Multi-Instance Multi-Label Image Classification	MMCNN-MIML	Put together CNN and multi-instance multi-label (MIML) learning for multi label image based classification.
Sun, 2018	Automatically designing CNN architectures using genetic algorithm for image classification	CNN	Proposed a way to automatically build an architecture of CNN for a particular image classification problem.
Gong, Zhong, Yu, Hu, & Li, 2019, p. 3615	A CNN with Multiscale Convolution and Diversified Metric for Hyperspectral Image Classification	CNN.	Proposed novel CNNs with multiscale convolution(MS-CNNs) to extract deep multiscale features

			from the hyperspectral image.
Zhao, Hao, He, Tang, & Wei, 2020, p. 264	A visual long-short-term memory based integrated CNN model for fabric defect image classification	VLSTM integrated CNN	Inspired by the human visual perception and visual memory mechanism, the paper proposed a visual-short-long-ter m memory (VLSTM) integrated CNN model for fabric defect image classification.
(Gour, Jain, & Sunil Kumar, 2020, p. 634)	Residual learning based CNN for breast cancer histopathological image classification	Residual Learning based CNN	Designed a residual learning based 152 layered convolutional neural network named ResHist for breast cancer histopathological image classification

Table 1 - Summary of major works

4. Methodology

The work is achieved through a classified four phases, i.e. data collection, pre-processing, training, and evaluation. Table 2 below presents the flow of each phase with its sub-tasks. The proposed system representation is shown in fig 1.

Overall Procedure

1.	**Dataset Collection**
2.	*Define all possible variation of photographs, documents and quote based images*
3.	*Gather images from possible sources*
4.	*Split the images into training and validation set*
5.	**Preprocessing**
6.	*Downscale images to 224*224 size.*
7.	*Normalise data based on the stats of RGB channels from the ImageNet dataset.*
8.	*Perform data augmentation.*
9.	**Training**
10.	*Perform Transfer Learning using ResNet34 model.*
12.	*Set dropout setting.*
13.	*Find the most appropriate learning rate for the model.*
14.	*Train the model.*
15.	**Evaluation**
16.	*Evaluate the model using test data.*
17.	*Calculate accuracy and top losses.*

Table 2 - A four phase approach to the proposed work

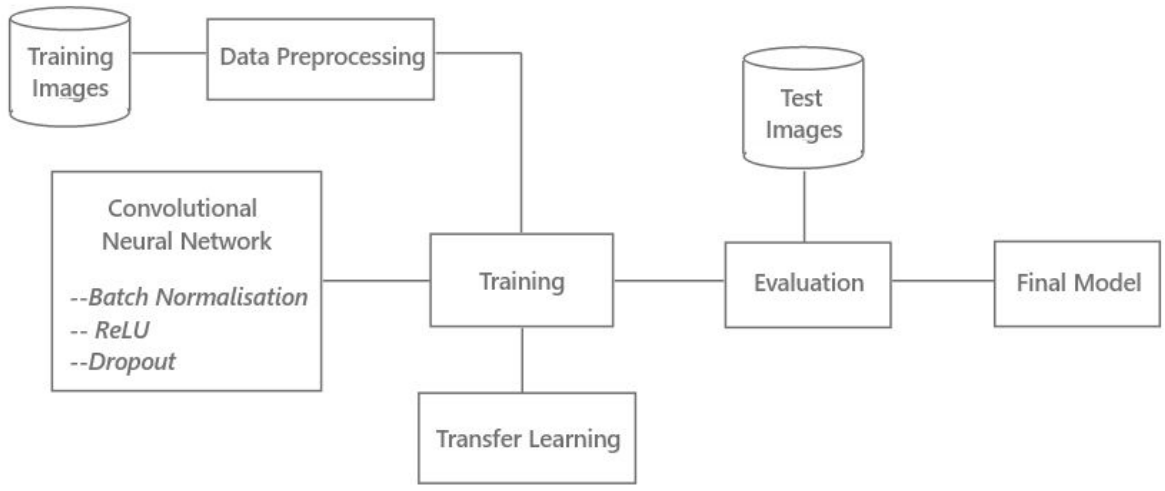

Figure 1 - The proposed system representation

4.1 CNN Network Architecture

The input for the network is a (3,224,224) image. In the first layer, 64 7x7 filters are used of stride (2,2) following with BatchNorm2D and ReLU. MaxPool2d with kernel size 3 and stride two is then used. Post that, there are multiple blocks of recurring Conv2D, BatchNorm2D, ReLu layers in the (a), (b), (c) sections of fig. 2. After this, a concatenation of both AdaptiveAvgPool2d and AdaptiveMaxPool2d is used followed by flattening of the last layer. BatchNorm1D layer and Dropout is used and a linear of size 512 is received. After using ReLU as activation function here and further using BatchNorm1D and dropout a linear of size 3 is received. Fig. 2 illustrates the CNN network architecture used in the classifier.

Figure 2: Architecture of CNN network

4.1.1 Transfer Learning

Transfer Learning is the method of reusing a model trained for one task on a second related task. Transfer Learning is very effective in the case of Deep Learning because an enormous amount of resources are required to train deep learning models. We are using a pretrained ResNet34 model which has been built on ImageNet. This helps us to use a portion of pre-trained weights which help to achieve great results with a small dataset.

4.1.2 Dropout

Proposed by Srivastava , dropout is one of the most effective ways of avoiding overfitting.In figure, the left side shows a fully connected layer where each line shows an activation times weight. In dropout, as inferred from figure 2, we randomly remove a few activations.

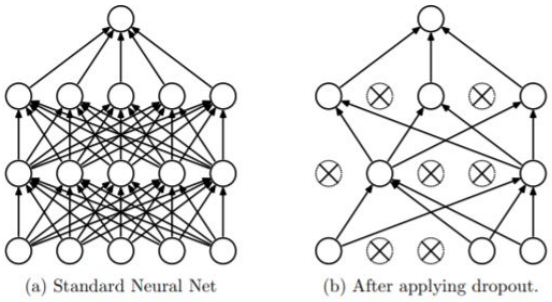

Figure 2: Image from (Srivastava, 2020) contrasting the difference between a standard neural network and neural network after applying dropout.

Usually, from within a mini batch that is going through, few activation at random are removed and then from the next, previous ones are kept and others are removed. This prevents the activation to learn some particular part of the input which in turn causes overfitting. If this overfitting happens, the model tends to learn an image in particular rather than learning features from it. In our case we have used a dropout rate of 0.5 for the fully connected layers.

4.1.3 Batch Normalisation

Since the time it was proposed by (Ioffe, 2015), Batch Normalisation has been used to improve accuracy and speed up training. Present at every layer, Batch Normalisation normalizes the hidden layers' activation values so that the distribution of the values remains the same during the training. The following steps are taken following the algorithm during the training process.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Figure 3: Batch Normalization Transform, applied to activation x over a mini-batch. Image from (Ioffe, 2015)

Mean is calculated of the activations followed by finding the variance. Post this, normalisation is done by subtracting the value and the mean and dividing it by the standard deviation. ε is added in the denominator to add numerical stability just in case σ^2 turns out to be 0 in some estimates. By the use of this, the speed of the training is significantly improved.

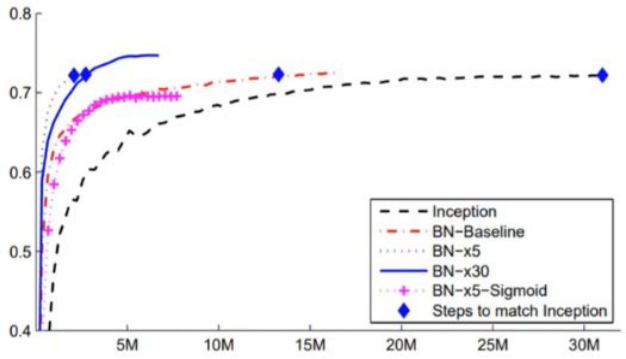

Figure 4: This image from (Ioffe, 2015) shows the improvement in speed by plotting single crop validation accuracy of Inception and its batch normalised variants vs number of training steps

4.1.4 Weight Decay

Weight Decay has been used as a type of regularisation. With a high number of parameters, the nonlinearity increases to a point of overfitting. Instead of reducing the number of parameters, their complexity is penalized. So a sum of squared parameters is added to the loss function. But

if this number is too large, the best loss would be shown if all the parameters were to be 0, which is no good. So this number is multiplied by a factor of *weight decay,* which is taken as .01.

4.2 Data collection and preprocessing

We created our own dedicated dataset from scratch by using Google images as our primary source. Approximately 800 images of the document, 350 images of photographs and 350 images of quotes were created. Almost all possible variations within a category were acknowledged. Document-based images include handwritten text, manuscripts, typed text, number sheets etc. Similarly, photograph images are a collection of selfies, group photographs, portraits etc and quote-based images include motivational quotes, birthday wishes, festival wishes etc. The high resolution images were downscaled to 224*224 size. About 20 percent images were used as validation set and normalisation of data was done based on the stats of RGB channels from the ImageNet dataset.

4.2.1 Data Augmentation

Data augmentation has been used for better generalization. We did not feed the model with the same picture every time, rather small random transformations such as a bit of rotation, zoom, flip etc were performed. These images might look the same to the human eye but have pixel values changed and help in generalizing the model better. This helps us to achieve better results in our relatively smaller dataset.

4.3 Training and Fine Tuning

4.3.1 Layer Deletion and Random Weights

We train our model on Resnet34 architecture with weights pre trained on the ImageNet architecture. The weight matrix in the end has a specific matrix of of 1000 columns, as this were the number of classification categories in the Image Net. We do not therefore require the same matrix in our case. So during the training process, the matrix is deleted and two new weight matrices are placed with ReLu in between instead, with the number of columns for the second matrix set as the number of classification categories i.e. 3 in our case.

These new matrices are filled with random numbers and are trained from scratch. The older layers however do not require training from scratch. (Zeiler, 2013) showed that the first layer was good at detecting diagonal edges. Layer 2 had a filter which was good at finding corners in the top left. As we move forward, we see that the more sophisticated and detailed detection is made. Therefore, we might want the weights used in the initial layers more than the last layers.

In the process, the initial layers were frozen, i.e. the backpropagation training was used only to update the parameters in the new layers and the parameters of the initial layers were kept untouched. This gives us faster and less memory consuming model along with utilization of weights that we need,

4.3.2 Cyclic Learning Rate and 1cycle policy

Smith, L. N. (2017) discovered a new method to set learning rate, called the Cyclical Learning Rate. In this method, the learning rate is allowed to oscillate between reasonable minimum and maximum bounds for a number of cycles with the optimal learning rate falling somewhere between the bound. This method might hinder the network performance temporarily, nevertheless in the longer run, it produces better results which are not computationally expensive

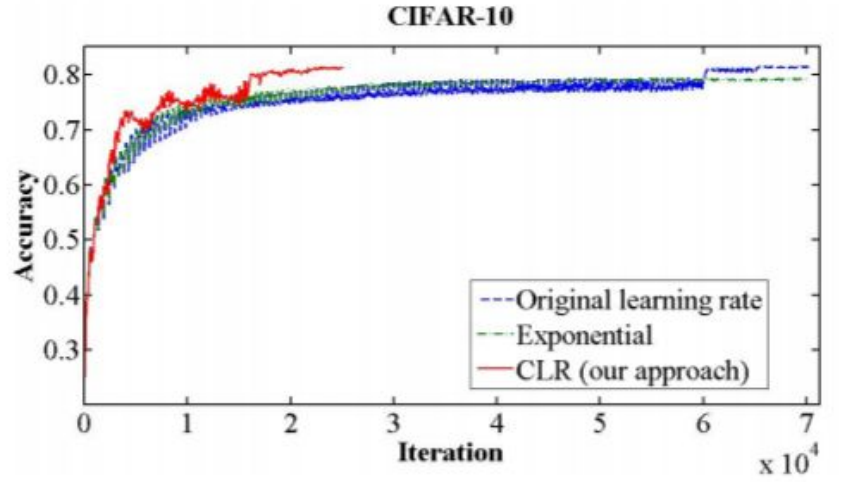

Figure 5: This image from Smith, L. N. (2017) shows the training accuracy of CIFAR-10 dataset over 70,000 iterations. A fixed learning learning rate (blue line) achieves 81.4% accuracy after 70,000 iterations, while the CLR method (red line) achieves the same within 25,000 iterations.

Building on the work of Cyclical Learning Rate, the author in Smith, L. N., & Topin, N. (2019) introduced super-convergence, a phenomenon by which a neural network can be trained an order of magnitude faster than with standard training methods. It uses the Cyclical Learning Rate method, but with only one cycle where the learning rate starts at a low value, increases to a very large value and then decreases to a value much lower than its initial value, hence gaining the name of 1cycle policy by the author. In our classification problem, we follow this 1cycle policy for setting our learning rate.

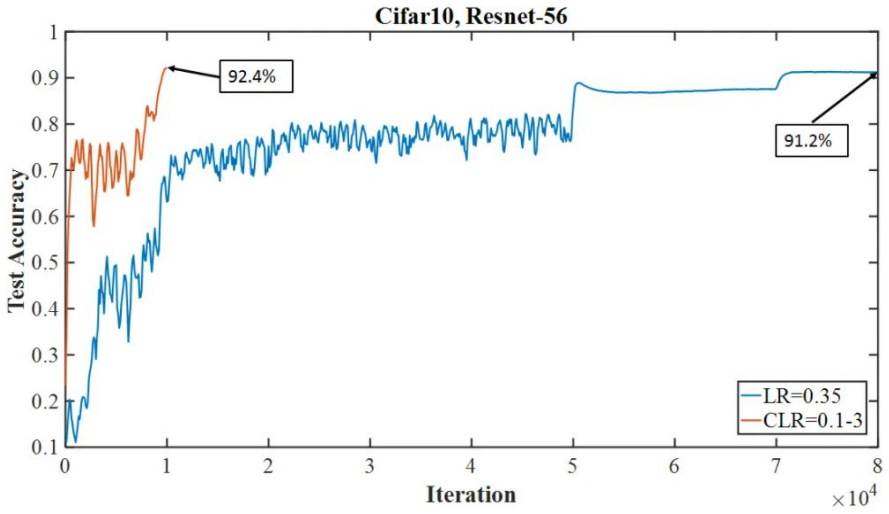

Figure 6: In this image from Smith, L. N., & Topin, N. (2019), we see that the super-convergence accuracy curve in red has a dramatic initial jump because of high learning rate, then oscillates or even declines for a bit due to the high learning rate and then jumps again to a distinctive accuracy peak because learning rate decreases to a very small value.

4.3.3 Discriminative Learning Rate

We now need to train the entire network after the parameters in the last layer have been updated reasonably well. So we unfreeze the layers, that is to say that parameters in all the layers will now be updated. But we know that the newer layers need more training than the previous layers with the initial layers requiring no learning at all. So different layers are divided into groups and each group is given a different learning rate. The learning rate of the initial layer groups is kept less as compared to the learning rate of the layer groups present at the end. This allows the values in the initial layers to not change a lot. In our scenario, three layer groups were used, with learning rates of 1e-5 for first, 1e-4 for middle and 1e-3 for the end.

5. Experimentation and Result

The model used 2016 images in the training set and 503 images in the validation set to be categorized within 3 classes.

As mentioned in methodology, because we are using pre trained weights from the ResNet34 model, we freeze the initial layers of our model and only train the weights in the new layers initially. While training we used the 1cycle policy to set the learning rate and achieve superconvergence instead of a fixed learning rate which gave us better results.

Epochs	Training loss	Validation loss	Accuracy	Precision	Recall	Fbeta
0	0.683233	0.298576	0.914513	0.914728	0.914476	0.914355
1	0.437397	0.212227	0.926441	0.926552	0.926661	0.926573
2	0.335256	0.223178	0.918489	0.917312	0.918648	0.918319
3	0.280439	0.221944	0.912525	0.911844	0.914158	0.913538

Table 3 - Classification result post initial training of the dataset

Epochs	Training loss	Validation loss	Accuracy	Precision	Recall	Fbeta
0	0.192738	0.198429	0.922465	0.922476	0.922733	0.922585
1	0.176548	0.225567	0.924453	0.923996	0.924578	0.924433
2	0.164426	0.190514	0.922465	0.922575	0.922013	0.922114
3	0.151066	0.191498	0.926441	0.927328	0.925139	0.925469

Table 4 - Classification result post initial training of the dataset using 1cycle policy

In table 2, the results of classification are shown that were achieved without using 1cycle policy to set the learning rate. It can be seen that the accuracy recorded after the 4 epochs is 91.25%. Training loss came down to 0.280 from 0.683 and validation loss started with approximately 0.298 and came down to 0.222 approximately.
Whereas, we can easily recognize the improvement in the results after using 1cycle policy to set the learning rate in table 3. We can see that the accuracy that was recorded in the initial epochs is higher than the accuracy post 4 epochs in table 2. An accuracy of 92.64% is achieved after the training. Similarly, training loss from the early epochs was recorded very low when compared to the training loss in table 2. A final training loss of approximately 0.151 was seen. Validation loss, too, improved significantly right from the initial epochs when compared to table 2, recording approximately 0.191 post 4 epochs.

Due to the freezing of initial layers; out of the total 21,814,083 parameters in our network, only 546,435 parameters are trained and updated. After this initial training of the new layers we now

allow training of all the parameters in our network and use discriminative learning rates as mentioned in the methodology. To use discriminative learning rates, we run a learning rate finder, by starting with a low number and rising up to a high number, and recording the loss at every iteration and plotting these losses with the learning rate to find the most appropriate values.

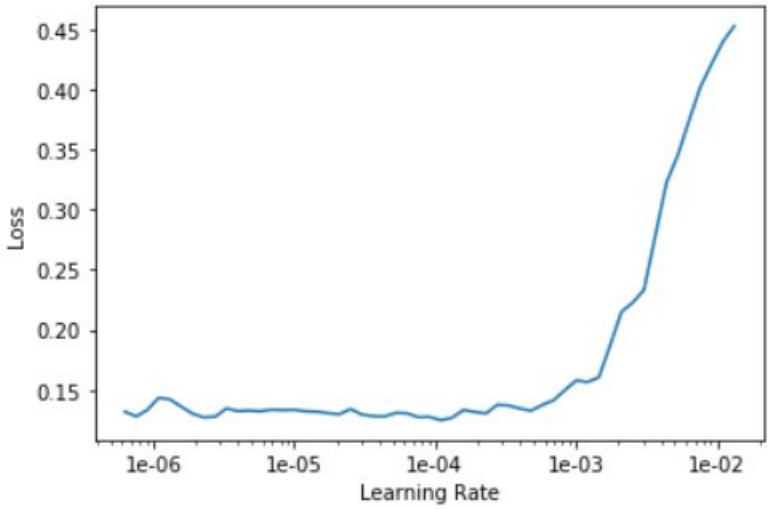

Figure 7: Plotting loss against the learning rates

We choose a range from the descent i.e. 1e-5,1e-3 for learning rates and use to train our 3 layer groups. A learning rate of 1e-5 is used for first, 1e-4 for middle and 1e-3 for the end. While training each of these layer groups, we here too follow the 1cycle policy for superconvergence.

Epochs	**Training loss**	**Validation loss**	**Accuracy**	**Precision**	**Recall**	**Fbeta**
0	0.197255	0.363730	0.892644	0.901915	0.894107	0.892930
1	0.226693	0.229555	0.918489	0.920502	0.920403	0.919748
2	0.189292	0.193526	0.938370	0.939942	0.938279	0.938350
3	0.135562	0.164980	0.948310	0.949758	0.947899	0.948108

Table 5 - Classification result post use of discriminative learning rates

Table 4 illustrates the result received after the use of discriminative learning rate. We see that our final model has achieved an accuracy of approximately 94.83% with 94.49% of precision, 95% approximately of recall and fbeta score. Table 5 illustrates in detail about the performance measures of the network. It can be clearly seen from fig 7 that the classifier is able to distinguish images between the three categories easily.

Category	**Precision**	**Recall**	**Fbeta**	**Support**
document_based	0.94	0.95	0.95	189
photographs	0.98	0.94	0.96	154
quote_based	0.93	0.96	0.94	160

Table 6 - Performance measure of the network

The confusion matrix for the classification is given in fig. 8. The highest error recorded was between classification of document-based images and quote-based images. This is evident because there are certain images that have bigger and fancy fonts and the classifier is not able to distinguish such images between documents and quotes. Similarly, in the case of classification of photographs and quote-based images, there are a number of images that have both photographs of people and a text associated with it, thus causing a discrepancy. The few images that weren't properly classified are the ones that are difficult to categorise even by a human-being because of a lack of a precise definition of boundaries between the three categories.

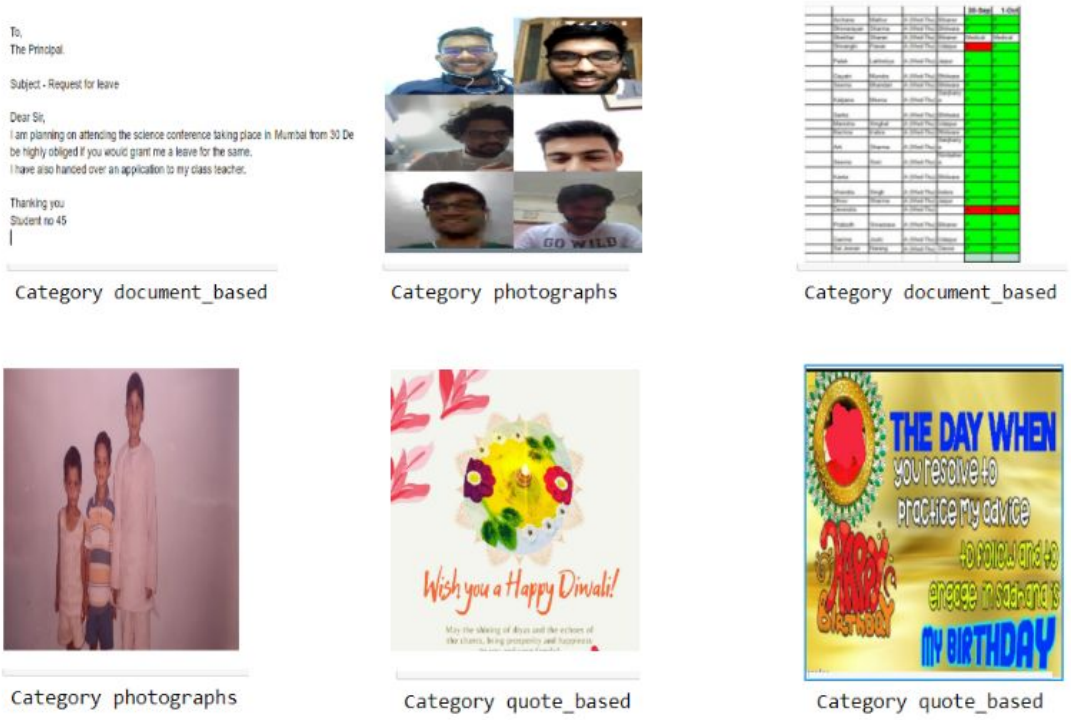

Figure 8 - Result: Social Media Image Classification

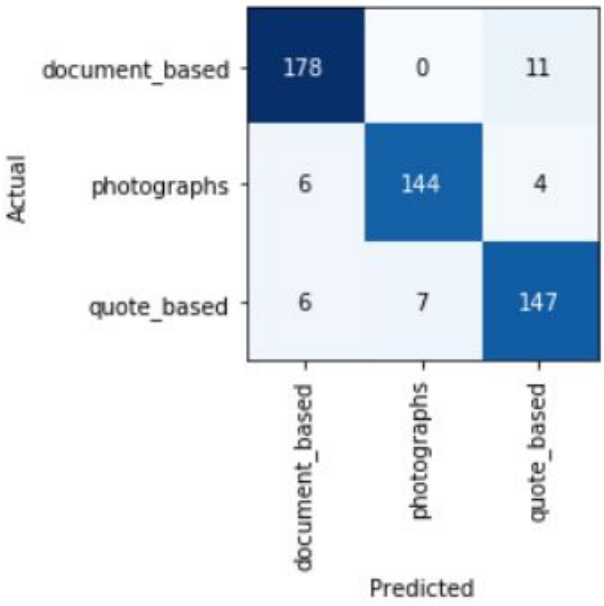

Figure 9 - Confusion Matrix

6. Conclusion

The objective of the paper was to build an image classification model to be used to classify images that are frequently collected in a device as a result of mass sharing of photographs. The goal was achieved successfully by using CNN to create an image classifier by using latest developments recently published in the field of CNN and deep learning in general that helped to achieve a very impressive accuracy. The work in this paper can be applied in mobile phones, tablets, laptops etc. in order to effectively manage abundant images that are stored in devices.This also opens up an opportunity to more closely analyse the photographs that are shared across social media platforms. If other broad categories of images appear in future or if particular categories of photographs are required to cater to the need of a specific problem, the same methodology can be used to train such categories of images.

7. References

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Zeiler, M. D. (2013, November 12). Visualizing and Understanding Convolutional Networks. Retrieved from https://arxiv.org/abs/1311.2901

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014). Medical image classification with convolutional neural network. *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, 844–848. https://doi.org/10.1109/icarcv.2014.7064414

Zha, S., Luisier, F., Andrews, W., Srivastava, N., & Salakhutdinov, R. (2015). Exploiting Image-trained CNN Architectures for Unconstrained Video Classification. *Procedings of the British Machine Vision Conference 2015*, 60.1-60.13. https://doi.org/10.5244/c.29.60

Liu, S., & Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 730–734. https://doi.org/10.1109/acpr.2015.7486599

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., & Quillen, D. (2017). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, *37*(4–5), 421–436. https://doi.org/10.1177/0278364917710318

Kolsch, A., Afzal, M. Z., Ebbecke, M., & Liwicki, M. (2017). Real-Time Document Image Classification Using Deep CNN and Extreme Learning Machines. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 1318–1323. https://doi.org/10.1109/icdar.2017.217

Tianyu, Z., Zhenjiang, M., & Jianhu, Z. (2018). Combining CNN with Hand-Crafted Features for Image Classification. 2018 14th IEEE International Conference on Signal Processing (ICSP), 554–557. https://doi.org/10.1109/icsp.2018.8652428

Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., & Atkinson, P. M. (2018). A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS Journal of Photogrammetry and Remote Sensing, 140, 133–144. https://doi.org/10.1016/j.isprsjprs.2017.07.014

Radenović, F. (2017, November 3). Fine-tuning CNN Image Retrieval with No Human Annotation. Retrieved from https://arxiv.org/abs/1711.02512

Chen, S.-W., & Tao, C.-S. (2018). PolSAR Image Classification Using Polarimetric-Feature-Driven Deep Convolutional Neural Network. *IEEE Geoscience and Remote Sensing Letters*, *15*(4), 627–631. https://doi.org/10.1109/lgrs.2018.2799877

Song, L., Liu, J., Qian, B., Sun, M., Yang, K., Sun, M., & Abbas, S. (2018). A Deep Multi-Modal CNN for Multi-Instance Multi-Label Image Classification. *IEEE Transactions on Image Processing*, *27*(12), 6025–6038. https://doi.org/10.1109/tip.2018.2864920

Sun, Y. (2018, August 11). Automatically designing CNN architectures using genetic algorithms for image classification. Retrieved from https://arxiv.org/abs/1808.03818

Ioffe, S. (2015, February 11). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Retrieved from https://arxiv.org/abs/1502.03167

Srivastava, N. (2020, June 18). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Smith, L. N., & Topin, N. (2019). Super-convergence: very fast training of neural networks using large learning rates. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 1. https://doi.org/10.1117/12.2520589

Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1. https://doi.org/10.1109/wacv.2017.58

Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., & Atkinson, P. M. (2018). A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *140*, 133–144. https://doi.org/10.1016/j.isprsjprs.2017.07.014

Gour, M., Jain, S., & Sunil Kumar, T. (2020). Residual learning based CNN for breast cancer histopathological image classification. *International Journal of Imaging Systems and Technology*, *30*(3), 621–635. https://doi.org/10.1002/ima.22403

Gong, Z., Zhong, P., Yu, Y., Hu, W., & Li, S. (2019). A CNN With Multiscale Convolution and Diversified Metric for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(6), 3599–3618. https://doi.org/10.1109/tgrs.2018.2886022