**PROJECT REPORT**

(Project Term January-May 2021)

**Image Classification using Pytorch CNN**

Submitted by:-

**Ansh Joshi Registration Number :11905461**

**Course Code INT246**

Under the Guidance of

**Dr. Sagar Pande**

# School of Computer Science and Engineering

A picture containing text

Description automatically generated

**DECLARATION**

We hereby declare that the project work entitled (“Image Classification using Pytorch CNN”) is an authentic record of our own work carried out as requirements of Project for the award of B.Tech degree in Computer Science from Lovely Professional University, Phagwara, under the guidance of Dr. Sagar Pande, during August to November 2020. All the information furnished in this project report is based on our own intensive work and is genuine.

Project Group Number: …………

Name of Student 1: Ansh Joshi

Registration Number: 11905461

Ansh Joshi

Date: 19 November 2021

**CERTIFICATE**

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfilment of the conditions for the award of B. Tech degree in Computer Science from Lovely Professional University, Phagwara.

**Dr. Sagar Pande**

**Designation**

**School of Computer Science and Engineering,**

Lovely Professional University,

Phagwara, Punjab.

Date : 19 November 2021

**ACKNOWLEDGEMENT**

Firstly, I would like to thank Almighty God, who have made me capable for completing this project in time. Then, I would like to thank Lovely Professional University, who have provided me with such an opportunity. Then, I would like to thank Dr. Sagar Pande sir, who have given such a wonderful topic, and who have helped me throughout my project, for the completion of my project. Then, I would like to thank my parents, who have provided me with such resources and capability that I can complete the project. I would also like to thank my friends, who have helped me whenever I felt like.

Thank you all……..

**TABLE OF CONTENTS**

Title Page……………………………………………………………………………(i)

Declaration…...………………………………………………………………………(ii)

Certificate……..…….………………………………………………………………(iii)

Acknowledgement…………………...………………………………………….......(iv)

Table of Contents……………………...……………………………………………(v)

**1. INTRODUCTION 1**

## 1.1. SECOND-LEVEL SUBHEADING 1

## 1.2. ANOTHER SECOND-LEVEL SUBHEADING 1

## 1.2.1. THIRD-LEVEL SUBHEADING 1

**1. INTRODUCTION**

Our CA of INT 246 was supposed to be assignment based, and we were required to submit a project of ML, AI by 20th November 2021. We were given a list of projects to choose from, but we were also given liberty to choose our own project. That’s when I thought of going for building a model for emotion detection using convolutional neural network, in Pytorch .

Then, I downloaded the dataset from Kaggle, consisting of 2 folders, each further consisting of 7 sub- folders. These 7 folders had many different images, each of different emotions, such as happy, disgusted, sad etc.

I had already done 1 course from Coursera, of machine learning and convolutional neural network, from Duke University. That was an amazing course, but it was long time ago, and I had never really practiced anything after that in convolutional neural network. That is why I thought I must go after this field and refresh my memories regarding this.

That course I did was using Pytorch library. Pytorch is a programming library built by Facebook. Pytorch is defined as an open-source machine learning library for Python. It is used for applications such as natural language processing. It is initially developed by Facebook artificial-intelligence research group, and Uber’s Pyro software for probabilistic programming which is built on it.

Originally, Pytorch was developed by Hugh Perkins as a Python wrapper for the LusJIT based on Torch framework. There are two Pytorch variants.

Pytorch redesigns and implements Torch in Python while sharing the same core C libraries for the backend code. PyTorch developers tuned this back-end code to run Python efficiently. They also kept the GPU based hardware acceleration as well as the extensibility features that made Lua-based Torch.

PyTorch is known for having three levels of abstraction as given below −

* Tensor − Imperative n-dimensional array which runs on GPU.
* Variable − Node in computational graph. This stores data and gradient.
* Module − Neural network layer which will store state or learnable weights.[1]

**2. PROFILE OF THE PROBLEM**

There are many different type of problems in machine learning. As we know that, machine learning is the process of training a piece of software, called model to make useful predictions using a data set. This predictive model can then serve up predictions about previously unseen data. We use these predictions to act in a product; for example, the system predicts that a user will like a certain video, so the system recommends that video to the user. But this act is not as simple as it seems. We encounter many problems while doing so. The most basic one being, exploring the data to get some useful insights at first hand to use it further. After that we choose a model and fit our data to that model to make predictions about new data.

But in convolutional neural network, the approach is a little bit different, as we cannot just choose any simple model for this. In convolutional neural network, we have images as input, and dealing with images is not that easy. Here, we have to convert our image to tensors, or as to say, 2d arrays, and then we run our kernel, or another 2d array over the image and then we make it learn how to predict true value, according to its answer and according to the label of the image. This kernel keeps on learning every time it runs over the dataset, and at the end it would have trained well enough to make good predictions for unknown data too.

Also, we also have to decide for how many times should we run our algorithm so that it does not overfit or underfit for our given dataset, both of which is not good for our model.

Another problem that we encounter is that deciding the accuracy of each model, which model works just fine for both testing and training dataset, because sometimes accuracy of one model for training set is too high, while for other set its too low, which means model is overfitted to training data.

**3. EXISTING SYSTEM**

Existing system is that we load necessary libraries into our model on our CPU, then we do some data pre-processing to visualize the data. After that we make our CNN model using PyTorch. Then, we initialize the model and run our model many times.

A tremendous interest in deep learning has emerged in recent years. The most established algorithm among various deep learning models is convolutional neural network (CNN), a class of artificial neural networks that has been a dominant method in computer vision tasks since the astonishing results were shared on the object recognition competition known as the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012.[2]

CNN is designed to learn spatial hierarchies of features automatically and adaptively through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. This review article offers a perspective on the basic concepts of CNN and its application to various radiological tasks and discusses its challenges and future directions in the field of radiology. Two challenges in applying CNN to radiological tasks, small dataset, and overfitting, will also be covered in this article, as well as techniques to minimize them. Being familiar with the concepts and advantages, as well as limitations, of CNN is essential to leverage its potential in diagnostic radiology, with the goal of augmenting the performance of radiologists and improving patient care.[2]

## 3.1 EXISTING SOFTWARE

I generally code in Anaconda python environment. It is famous tool for making python programs. Also, downloading libraries in anaconda is easy. Further, you can execute every block of code individually to check if there is any error or not. You do not have to run whole program at once.

It is a distribution of the python and R programming languages for scientific computing, i.e., data science, machine learning, large scale data processing, predictive analysis, that aims to simplify package management and deployment.

## 3.1 DFD FOR PRESENT SYSTEM

Here is the DFD for the present machine learning system that most programmers use: -

Diagram

Description automatically generated[3]

They divide the data into 2 parts, i.e., training and testing. They train the algorithm on training dataset till some good accuracy is achieved. Then, they test it on testing dataset, and if the model is found to be satisfactory, they deploy it.

**4. PROBLEM ANALYSIS**

## 4.1 PRODUCT DEFINITION

Speaking in simple language, machine learning is the study of analyzing the data and training our machine on that data well enough, so that machine can predict correctly what is desired from user, by using many algorithms.

And, in convolutional neural network we deal with images, i.e., we will iterate through all the images, convert these images to tensor and the kernel learns the pattern between them and learns how to predict next image correctly. Then if on iterating, our accuracy received is sufficient, we iterate it through our testing dataset images. And if then also our accuracy is good.

A convolutional neural network (CNN) is a type of artificial neural network and is used in image processing and processing that is specifically designed to process pixel data[4].

CNNs are powerful image processing, artificial intelligence that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).

A neural network is a system of hardware and/or software patterned after the operation of neurons in the human brain. Traditional neural networks are not ideal for image processing and must be fed images in reduced-resolution pieces. CNN have their “neurons” arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals. The layers of neurons are arranged in such a way as to cover the entire visual field avoiding the piecemeal image processing problem of traditional neural networks.[4]

**4.2 FEASIBILITY ANALYSIS**

Machine learning is a practice where computers use different algorithms to search relationships, patterns, and trends. The goal is to learn something new and make predictions about possible outcomes.

Machine learning is the future of industry, helping to achieve cost saving and revenue growth. Industrial companies that implement machine learning today will definitely achieve a competitive advantage.

As such, there is no such direct way to do feasibility analysis. The most trivial way is just to try it and see if it works, because many a times, data will be redundant, or datasets will be less in number. Hence, we are never sure whether it will be successful or not unless we try it.

For this feasibility study, following will be high level steps :

* For each feature, perform PCA with rest of the features as x\_train and feature as y\_train. If you find a feature that can be predicted with other features, ML can be applied on the dataset.
* Can human solve it? As a person, can you find patterns for a given feature based on other features.
* Perform exploratory data analysis (EDA) with the dataset and try it.[5]

Also, we should not get discouraged when we receive a bit low accuracy in our model building, because it may pretty much happen when our data relationship is not linear.

**5. SOFTWARE REQUIREMENT ANALYSIS**

**5.1 INTRODUCTION**

Basically, software requirement analysis means complete study analysing, describing software requirements so that requirements that are genuine and needed and can be fulfilled to solve the problem.

Graphical user interface, application

Description automatically generated[6]

**5.2 GENERAL REQUIREMENT**

Software requirement for machine learning and CNN include some characteristics such as fast processing, because we will be requiring to iterate over a large amount of data. We may need at least 4 GB of RAM, if not more, because otherwise computer may lag.

**5.3 SPECIFIC REQUIREMENT**

We may even require Graphical Processing Unit (GPU) for image processing and deep learning, but it is not necessary. A GPU is a mini version of an entire computer but only dedicated to a specific task. It is unlike a CPU that carries out multiple tasks at the same time. GPU comes with its own processor which is embedded onto its own motherboard coupled with v-ram or video ram, and also a proper thermal design for ventilation and cooling.

One of the most admired characteristics of a GPU is the ability to compute processes in parallel. This is the point where the concept of **parallel computing** kicks in. A CPU in general completes its task in a sequential manner. A CPU can be divided into cores and each core takes up one task at a time. Suppose if a CPU has 2 cores. Then two different task’s processes can run on these two cores thereby achieving multitasking.

But still, these processes execute in a serial fashion.

GPU will be having hundreds and thousands of cores, all of which are dedicated towards a single task. These are simple computations that are performed more frequently and are independent of each other. And both store frequently required data into their respective cache memory, thereby following the principle of locality reference.

Though, it does not mean that we necessarily require GPU for machine learning. We actually just need a IDE and an interpreter which can run our code, that would work too.[7]

**6. DESIGN**

While designing the code for my project, I took a lot of help from online resources, from PyTorch official website and many articles. But the idea was my original, and code was my original, i.e., I did not copy pasted it from any website, I just took help from them.

**6.1 SYSTEM DESIGN**

Diagram

Description automatically generated

This is the design which every machine learning programmer follows while making CNN model. We train our sample data on our model consisting of convolutional layer, linear layer, pooling layer, batch normalizing. After building the model we test it on our test set.

**6.2 PSEUDO CODE**

Step 1: - Import necessary libraries

Step 2: - Set the path for training dataset

Step 3: - Visualize the data of every folder

Step 4: - Use ImageFolder for defining transforms, i.e., converting images to tensor.

Step 5: - Use DataLoader for spiting out random samples of our data of certain batch size as we wish

Step 6: - Visualize any random image from our batch just to make sure that we have got tensors right.

Step 7: - Make the class for classification consisting of as many convolutional, linear, pooling layers as you deem necessary.

Step 8: - Initialize the model and define criterion and optimizer.

Step 9: - Run the model for many epochs, as you like.

Step 10: - Calculate the accuracy for our dataset.

Step 11: - Do the steps 4-5 and 9-10 for testing data. And calculate its accuracy too

Step 12: - If accuracy of both is same and quite high then proceed, otherwise repeat from step 4.

Step 13: - That’s it, now our model is ready to be deployed to real world.

**7. TESTING**

For testing our model, we already had another folder of images with unknown labels, and after our model had sufficient accuracy, we iterated our model over those images, and we predicted the accuracy.

Even for training our model, we have different accuracy metrics such as cross entropy loss, mean squared error, maximum likelihood etc. For regression problems, we use criterions as mean squared error, and for classification problems, such as ours, we use cross entropy and logarithmic loss.[8]

After this we get to testing data. Training data is different from testing data. It is also important to note that no value of training data should be in testing data. Otherwise, our model will not perform well in real world when it would be employed, because it will be containing bias towards that data value.

Testing is necessary for the model because we may never know if the model is good enough until we try it to execute some unknown data and see its performance.

I just iterated my model over that folder containing images, because I did not have any experience of other type of testing available to us.

**8. IMPLEMENTATION**

**8.1 IMPLEMENTATION OF THE PROJECT: -**

Implementing a CNN machine learning model itself is a difficult process.

For example, if you wish to implement it in websites and web applications, you may require Flask. Following are the steps for implementing it in website:-

1. Create your machine learning model
2. Develop your web application with Flask and integrate your model in the app
3. Deploy your web-app in Heroku Cloud Platform

For using machine learning in an android app, if you are a beginner, you may require ML kit for android and Core ML for Apple. If you are a full developer, you may require Amazon’s SageMaker which is cloud-based service provided by Amazon, or Google ML engine which is cloud-based service by Google. You can even go for Microsoft azure AI or IBM Watson.

Given below is the difference between all these cloud services, you can compare them all and choose which fits your criterion.

A picture containing graphical user interface

Description automatically generated[9]

**8.2 POST IMPLEMENTATION AND SOFTWARE MAINTAINENANCE**

Regarding post implementation and software maintenance, I will keep on trying different ways of hyperparameter tuning of different hyperparameters, so as to increase my model’s accuracy. I may actually add more layers, if needed.

I will also add more images for the model, if needed. Then, model will be able to learn from data better.

**9. USER MANUAL**

**FOR THE CODE GIVEN ON NEXT PAGES**

1. path refers to the path where training images will be stored.
2. target\_var refers to list of folders of the train set we obtained from os.listdir.
3. x is a temporary variable used while visualizing the images.
4. train\_set refers to the path where images, after being converted to tensors are stored.
5. train\_set is again used to overwrite original value with small batches .
6. dataiter is an iterator through our batch.
7. imcl is the name of class we have used for our model.
8. x in def forward is the way our model will execute given upper layers.
9. model is how we initialized our class
10. criterion and optimizer are the criteria of our loss function and how we will optimize our model respectively.
11. Epoch is number of times our code will run.

**10. SOURCE CODE**

**(SNAPSHOTS)**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, email

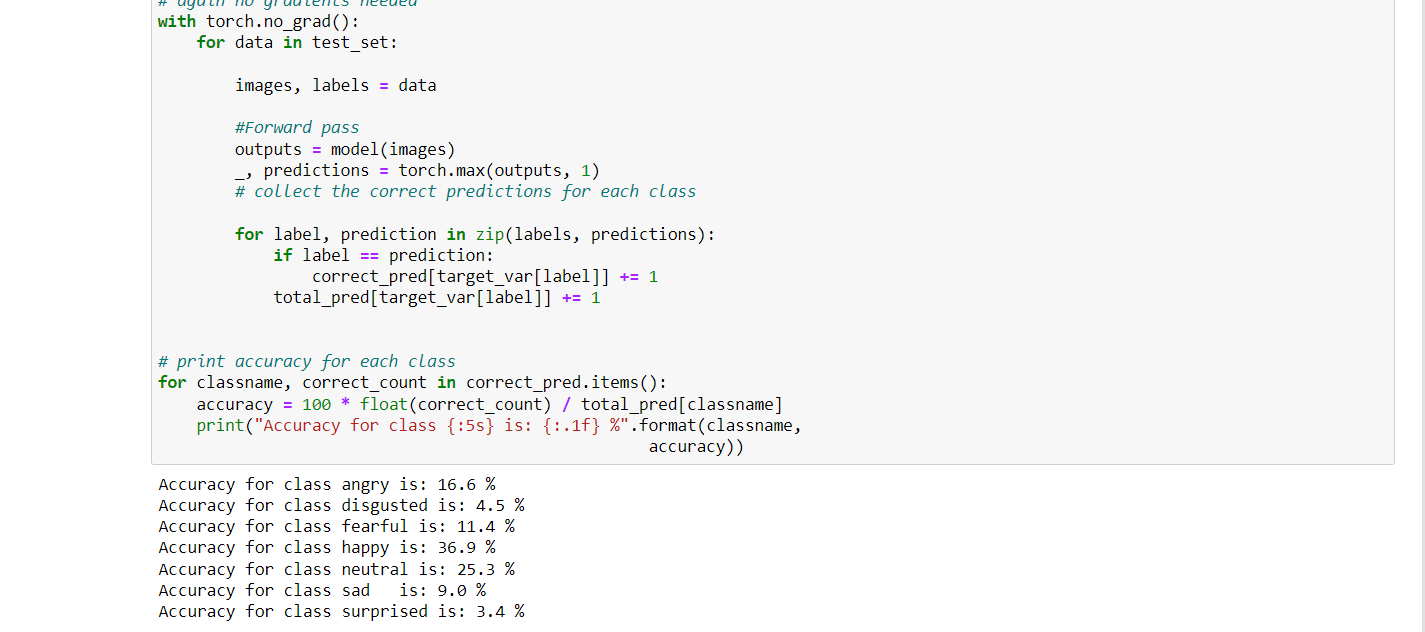
Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



**CITATIONS**

[1] - <https://www.tutorialspoint.com/pytorch/pytorch_introduction.htm>

[2]- <https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9>

[3]- <https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94>

[4]- <https://searchenterpriseai.techtarget.com/definition/convolutional-neural-network#:~:text=A%20convolutional%20neural%20network%20(CNN,designed%20to%20process%20pixel%20data.&text=A%20neural%20network%20is%20a,neurons%20in%20the%20human%20brain>.

[5] - <https://towardsdatascience.com/assessing-the-feasibility-of-a-machine-learning-model-ae36f4180f8>

[6] - <https://data-flair.training/blogs/machine-learning-software/>

[7] - <https://www.analyticsvidhya.com/blog/2020/09/why-gpus-are-more-suited-for-deep-learning/>

[8] - <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>

[9] - <https://www.kdnuggets.com/2018/01/mlaas-amazon-microsoft-azure-google-cloud-ai.html>

**BIBLIOGRAPHY**

For making my project, I took help from many online resources, such as these: -

* <https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html>
* <https://discuss.pytorch.org/>
* <https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/>
* <https://www.pyimagesearch.com/2021/07/19/pytorch-training-your-first-convolutional-neural-network-cnn/>
* <https://medium.com/thecyphy/train-cnn-model-with-pytorch-21dafb918f48>