Nosql Project

on

Image Processing using Transfer Learning With Neo4j

Submitted By:-

Nayan Verma – 17U03026

Submittd to:-

Dr. Robin Singh Bhadoria

Contents

[1. Introduction 2](#_Toc5976695)

[2. Related Work 3](#_Toc5976696)

[2.1. Concept Existing Model 3](#_Toc5976697)

[2.2. Benefits 3](#_Toc5976698)

[2.3. Drawbacks 4](#_Toc5976699)

[3. Implementaiton 4](#_Toc5976700)

[3.1. Code 4](#_Toc5976701)

[3.2. Programming Language 7](#_Toc5976702)

[3.3. Backend 8](#_Toc5976703)

[4. Project Contribution 9](#_Toc5976704)

[4.1. Idea 9](#_Toc5976705)

[4.2. Motivation 9](#_Toc5976706)

[5. Conclusion 10](#_Toc5976707)

[Works Cite 11](#_Toc5976708)

# Introduction

This project uses a trained model named “Mobilenet” , extracts it till last layer(with the help of ml5.js library) and then the model is trained with a set of images that we provide , this method is knows as transfer learning(feature-representation-transfer) .

The images used to train the model are stored in the hard drive, there location and label associated with it is stored in the nosql database software named Neo4j with the type of image captured. I have also included node.js which allows socketing.

The user sees a live web cam video on their web browser, there is a text field and a button bellow it. The user can train the model by telling it the kind of image it is looking at and help it identify other images that is sees. The predications are displayed on the video itself.

# Related Work

## Concept Existing Model

ML5.js is a library in JavaScript that is built on top of tensorflow.js which provide machine learning classes. This library allows us to extract the features of a pre-trained model named “Mobilenet”.

It also allows creation of KNN-classification object, but there is no method to save the trained model. You can download a JSON file and load it again when the program restarts. The web browser will download JSON file.

This method has faults, what if the program fails in between or the machine faults , there is no way to recover the trainings of that model.There is no way to know what images are used to train the model.

## Benefits

The data set stored to train the model can be used to train other models. Neo4j keeps the record of the images and all the labels it is associated with, different type of images can be used to train different models.

The database is updated at the time of training the model when the KNN-Classifier object is given with the logits and label that the logits belong to, so if there is a fault during the execution of the program, the model can be trained again with the set to images stored

The images are stored there labels are stored, there relations with different nodes is saved so if some images are to be removed , they easily can be.

Neo4j is Schema-free and schema-optional, Flexible which is essential in the project, the relationship that images have with each other images must be stored and neo4j, a graph based nosql database, is ideal to that.

## Drawbacks

The images stored creates a burden to the storage. As the images increase the loading time increases. We need a high end machine to run the program.

# Implementaiton

## Code

let features;

let video;

let knn;

let ready=false;

let label='Wait';

let imageNumber=0;

function preload()

{

video=createCapture(VIDEO);

video.hide();

knn=ml5.KNNClassifier();

features= ml5.featureExtractor('MobileNet',modelReady); //this is the model extractedS

}

function function1(i,node){

let FIELD=node.\_fields[0];

let img=createImg("Images\\"+FIELD.properties.URL, function(){

var logits=features.infer(img);

knn.addExample(logits, FIELD.properties.Label );

imageNumber++;

}).hide();

}

function modelReady()

{

console.log('Model is Ready');

const driver = neo4j.v1.driver("bolt://localhost", neo4j.v1.auth.basic("neo4j", "1234"));

const session = driver.session();

const resultPromise = session.writeTransaction(tx => tx.run(

'Match(n) return n;') );

resultPromise.then(result => {

session.close();

console.log('\*\*');

const node=result.records;

for( i=0;i<node.length;i++)

{

function1(i,node[i]);

/\*FIELD=node[i].\_fields[0];

img=createImg("Images\\"+FIELD.properties.URL, function(){

var logits=features.infer(img);

knn.addExample(logits, FIELD.properties.Label );

imageNumber++;

}).hide();

}

// on application exit;

driver.close();

}).then(function (){console.log('PreLoad complete'); ready=true;}).catch(err=>{console.log(err)});

}

function videoReady()

{

console.log("Video is Ready");

}

function saveTheImage()

{

saveCanvas(imageNumber+".jpg");

const driver = neo4j.v1.driver("bolt://localhost", neo4j.v1.auth.basic("neo4j", "1234"));

const session = driver.session();

const resultPromise = session.writeTransaction(tx => tx.run(

'CREATE(n:Image{Label : $L , URL: $U})', {L : input.value(), U : imageNumber+'.jpg'}) );

resultPromise.then(result => {

session.close();

console.log('\*\*');

// on application exit;

driver.close();

}).catch(err=>{console.log(err)});

imageNumber=imageNumber+1;

}

function setup() {

// put setup code here

createCanvas(640, 480);

background(0);

input=createInput();

input.position(20,height+40);

butt=createButton("Train Machine");

butt.position(20+input.width,height+40)

butt.mousePressed(function ()

{

var logits=features.infer(video);

saveTheImage();

knn.addExample(logits, input.value()); ///\*\*////

console.log(logits);

});

}

function doTheClassification()

{

const logits=features.infer(video);

knn.classify(logits,function(error,results){

label=results.label;

doTheClassification();

})

}

function draw() {

// put drawing code here

background(200,200,200);

image(video,0,0,video.width,video.height);

if(ready && knn.getNumLabels()>0)

{

doTheClassification();

ready=false;

}

if (label != '') {

push();

textSize(26);

frameRate(10);

fill(random(255),random(255),random(255));

//rotate(random(2\*PI));2

text(label, 0,470);

pop();

}

}

This code is written for p5.js.

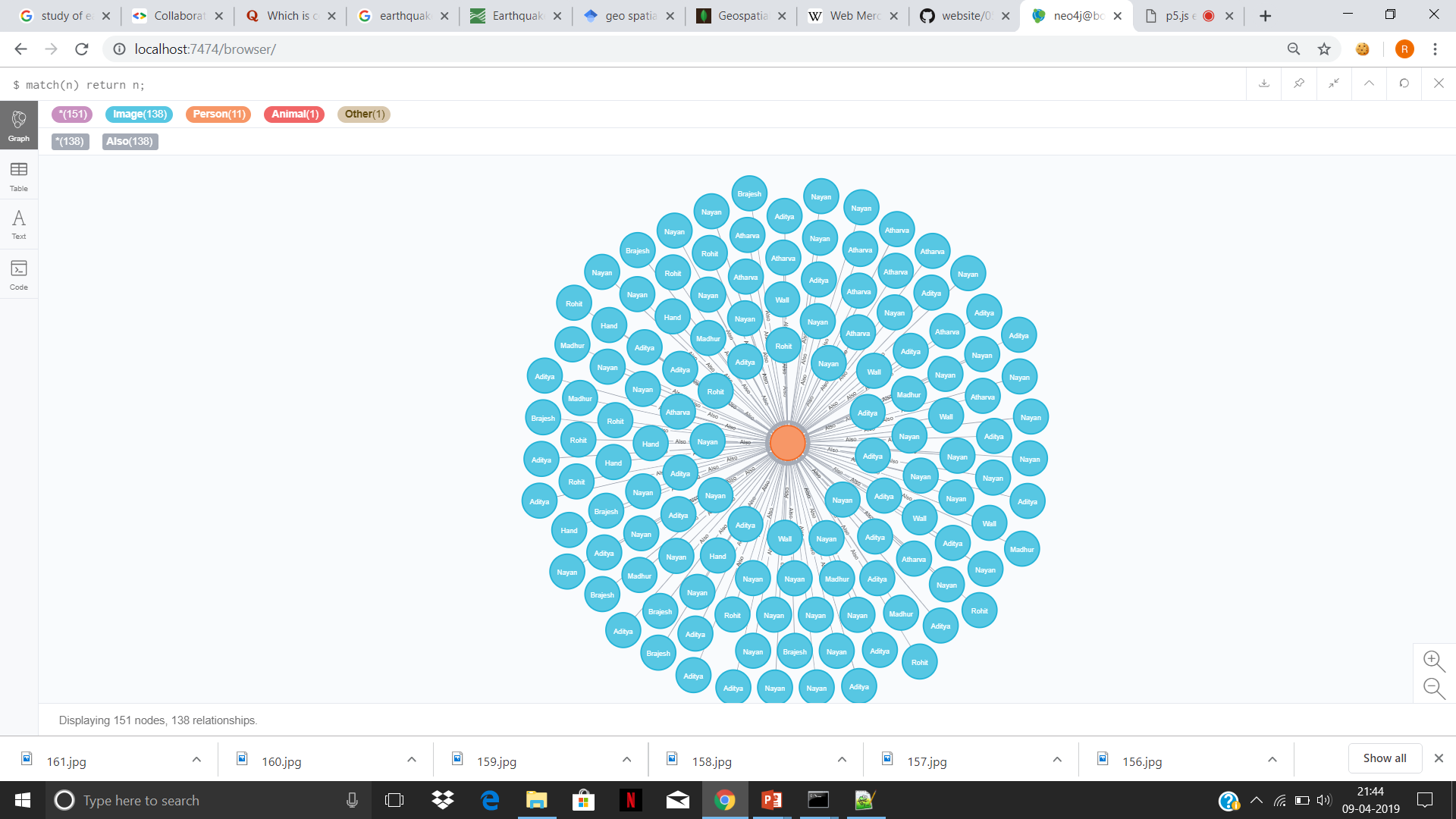
## Programming Language

JavaScript (JS) is a lightweight interpreted or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

This project uses the P5.js , ML5.js and Node.js libraries of JavaScript. The p5.js help in the UI. It contains functions to display the web-cam video to the web page, all the inputs and buttons. The ML5.js library helps in extracting the features of Mobilenet, and make the object of KNN-Classifier.Node.js helps in socketing, so that multiple users can use the program.

## Backend

Neo4j graph platform takes a connections-first approach to data. It broadens a company’s ability recognize the importance of persisting relationships and connections through every transition of existence: from idea, to design in a logical model, to implementation in a physical model, to operation using a query language and to persistence within a scalable, reliable database system. The foundation of representing connected data is known as a graph.



# Project Contribution

## Idea

The idea of this project is to implement transfer learning using neo4j to store the images that the model is to be trained for. The idea to provide an alternative way to the existing transfer learning technique to train a model. Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

## Motivation

A software that that recognize images with the label that we provide, to implement transfer learning and store the images so that the model can be trained later and the images can be used to train other models. Inductive transfer learning approach is applied. where the source data domain is same as target data domain but the source task is different from target task.

# Conclusion

It was a pleasure that I got an opportunity to make this project .

This project uses a trained model named “Mobilenet” , extracts it till last layer(with the help of ml5.js library) and then the model is trained with a set of images that we provide , this method is knows as transfer learning(feature-representation-transfer) .

The images used to train the model are stored in the hard drive, there location and label associated with it is stored in the nosql database software named Neo4j with the type of image captured. I have also included node.js which allows socketing.

The user sees a live web cam video on their web browser, there is a text field and a button bellow it. The user can train the model by telling it the kind of image it is looking at and help it identify other images that is sees. The predications are displayed on the video itself.

# Works Cite

[1] Fiebrink, R. (2019). Machine Learning Education for Artists, Musicians, and Other Creative Practitioners. *ACM Transactions on Computing Education*.

[2] Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007, June). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning* (pp. 759-766). ACM.

[3] Torrey, L. and Shavlik, J., 2010. Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI Global.

[4] Torrey, L. and Shavlik, J., 2010. Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI Global.