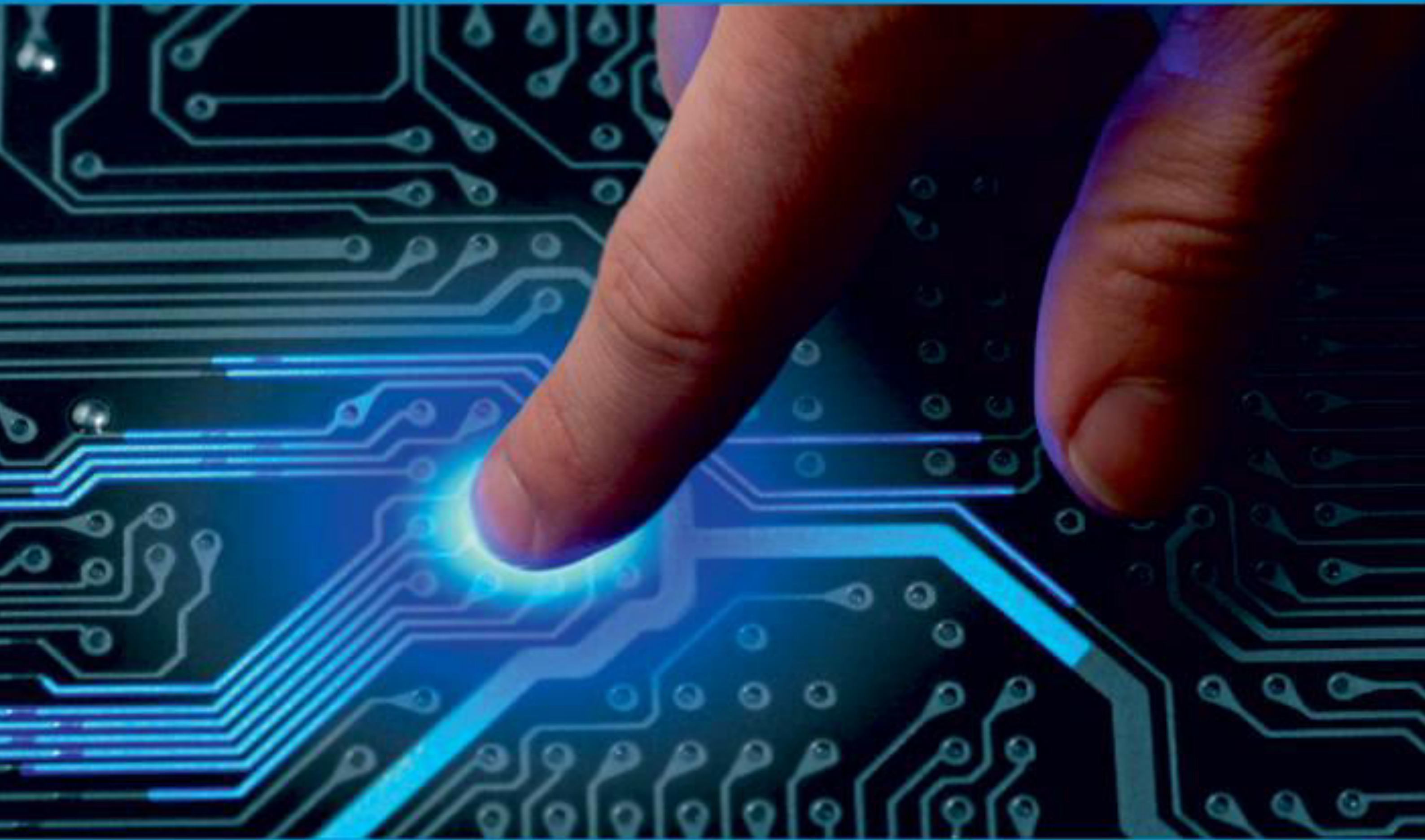
e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 12, December 2024

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 12, December 2024| DOI: 10.15680/IJIRCCE.2024.1212092

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Fruits Image Classification Using Deep Learning

Sunitha M, Banushri S

PG Student, Department of C.S.E., ICEAS, Vishweshwaraya Technological University, Bangalore, India

Assistant Professor, Department of C.S.E., ICEAS, Vishweshwaraya Technological University, Bangalore, India

ABSTRACT: The aim of this proposed work is to build a fast and accurate fruit detection system. The proposed project can detect some fruits. This helps us to gain some extra Knowledge. This is mainly useful for fruit Industries to allocate the fruit in a particular box. Fruit detection is used to reduce the effort of humans. To identify the fruit types used on artificial neural networks. It is the best model for the deep learning technique to easily identify the fruit types. Deep Learning models, or to be more specific, TensorFlow uses Convolutional Neural Networks (CNN) models are proposed in this project. A comparison is done between proposed algorithms and current algorithms reveals that the accuracy of fruits detection using CNN is greater than other algorithms. Thus higher accuracy is achieved and able to detect fruits and deploying the model in a android studio using Tensorflow Framework. Image classification is used to close the visual perception gap between computers and humans, enabling machines to recognise images in the same manner that people do.

KEYWORDS: Convolutional Neural Networks; TensorFlow; Keras; network lifetime; Dataset; Android Studio; Deep Learning

I. INTRODUCTION

In Deep Learning, a convolutional neural network(CNN/ConvNet) it is a class of deep neural network, mostly applied for visual imaginary. Learning can be supervised,semi-supervised, or unsupervised. For the second objective, we have trained a Convolutional Neural Network Model which is capable of identifying fruits.

Recognizing different kinds of fruits is a difficult task in supermarkets, since the cashier must point out the categories of a particular fruit to determine its price. The use of barcodes has mostly ended this problem for packaged products but given that most consumers want to pick their products, they cannot be pre-packaged, and thus must be weighed. A solution is issuing codes for every fruit, but the memorization is problematic leading to pricing errors. Another solution is to issue the cashier an inventory with pictures and codes, however, flipping over the booklet is time consuming. Automatic classification of fruits via computer vision is still a complicated task due to the various properties of many types of fruits. The fruit image detection technique which was based on external properties of fruits such as shape, size and color.

Currently, the machine learning model is applied in many fields such as facial recognition, handwriting processing, or object tracking . One of the growing interesting areas is object classification. Although it is only a small subset of computer vision, its range of applications is huge. It underpins a lot of important applications. Therefore, we propose to build a fruit classification system using Tensorflow model in the paper. The simulation results show that the model has high accuracy and can be applied for real environments. In the paper, there are three points that we propose as follows. Firstly, we change the network architecture according to the requirements. Secondly, we propose the fruit classification system using Tensorflow model. Finally, we modify dataset that are suitable for real applications. As a result, accuracy of proposal model is improved up to 99%.

II. RELATED WORK

This reports a four-beam 1 GHz/2.3 GHz dual frequency sensor system to assess the quality of thick rind pomelo fruits. The principle of the proposed sensor system is based on phase difference between lower- and higherfrequency reflection coefficients. The sensor system consists of five dual-band antennas and a customized circuit to approximate. The main components of the customized circuit are down conversion mixers and phase detector modules. Growing global consumption of thick rind fruits has contributed to their greater economic significance. There are a variety of

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

thick rind fruits, such as pomelo, watermelon, and durian. To command premium prices, quality control of the fruits is paramount. Conventionally, fruit quality assessment relies on human sensory input. However, the practice is subjective, leading to inconsistent fruit quality and reputation damage. Thick rind fruits, e.g., pomelo, watermelon, are typically of spherical shape or, specifically, concentric dielectric spheres with rind as the outer layer and flesh as the inner layer. In this research, simulations were carried out using a pomelo fruit model whose outer and inner diameters are 16 and 12 cm, which are the typical size of pomelo fruit. The flesh portion of a typical pomelo fruit consists of 15 sections, and the simulations are to differentiate between normal and granulated pomelo fruits. Granulation is a condition in which juice vesicles become dry and tough, which makes the fruit unpalatable. In the simulation, the pomelo that contains one granulated section (out of 15 sections) is regarded as failing. Prior to simulation, the relative permittivity of real pomelo fruits (three each for normal and granulated pomelos) were determined using Agilent 85070E dielectric probe connected to N9916A Field Fox microwave vector network analyzer (Figure 3(a)). The relative permittivity of the rind and flesh was measured using the dielectric probe, To obtain accurate relative permittivity, the pomelo fleshes were homogeneously pressed prior to measurement by the dielectric probe. Simulations were carried out using single- (1 GHz) and dual-frequency (1 GHz/2.3 GHz) sensor systems; and the results indicated that the singlefrequency scheme achieved lower accuracy in classification of pomelo fruits. As a result, a prototype of a four beam dual-frequency sensor system was fabricated, and experiments were carried out using plastic pomelo models and real pomelo fruits. They further demonstrate The experimental results showed that the average of the normal and granulated pomelo fruits are almost identical, while SD of the granulated pomelo is noticeably larger than that of the normal pomelo. As such, SD is used as the determinant of pomelo quality.

III. PROPOSED ALGORITHM

A. *Design Considerations:*

Below is a detailed step by step process for implementing image classification based on Convolutional Neural Networks(CNNs) using the TensorFlow framework.

- Examine and understand data
- Build an input pipeline
- Build the model
- Train the model
- Test the model
- Improve the model and repeat the process

B. *Description of the Proposed Algorithm:*

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve stateof-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

In a word, accuracy. Deep learning achieves recognition accuracy at higher levels than ever before. This helps consumer electronics meet user expectations, and it is crucial for safety-critical applications like driverless cars. Recent advances in deep learning have improved to the point where deep learning outperforms humans in some tasks like classifying objects in images.

1. Deep learning requires large amounts of labeled data. driverless car development requires millions of images and thousands of hours of video.
2. Deep learning requires substantial computing power. High-performance GPUs have a parallel architecture that is efficient for deep learning. When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks.

The term "deep" usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150. Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction

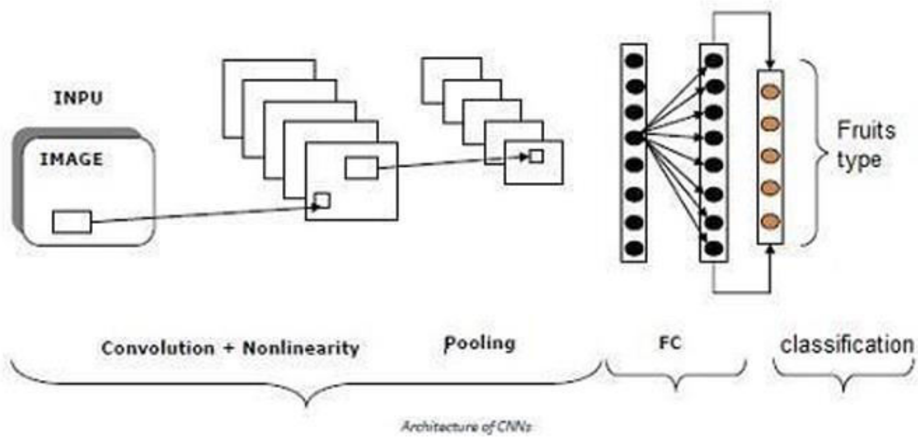


Fig.1.Convolutional neural network

One of the most popular types of deep neural networks is known as convolutional neural networks (CNN or ConvNet). A CNN convolves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images.

CNNs eliminate the need for manual feature extraction , so you do not need to identify features used to classify images. CNN works by extracting features directly from images. The relevant features are not retrained; they are learned while the network trains on a collection of images. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification.

CNNs learn to detect different features of an image using tens or hundreds of hidden layers. Every hidden layer increases the complexity of the learned image features. For example, the first hidden layer could learn how to detect edges, and the last learns how to detect more complex shapes specifically catered to the shape of the object we are trying to recognize.

IV. PSEUDO CODE

```
package com.example.myapplication;

import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.media.ThumbnailUtils;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
```

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

```
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.example.myapplication.ml.Model;
import org.tensorflow.lite.DataType;
import org.tensorflow.lite.support.tensorbuffer.TensorBuffer;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
public class MainActivity extends AppCompatActivity {

    Button camera, gallery;
    ImageView imageView;
    TextView result;
    int imageSize = 32;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        camera = findViewById(R.id.button);
        gallery = findViewById(R.id.button2);

        result = findViewById(R.id.result);
        imageView = findViewById(R.id.imageView);

        camera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (checkSelfPermission(Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
                    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                    startActivityForResult(cameraIntent, 3);
                } else {
                    requestPermissions(new String[]{Manifest.permission.CAMERA}, 100);
                }
            }
        });
        gallery.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent cameraIntent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                startActivityForResult(cameraIntent, 1);
            }
        });
    }

    public void classifyImage(Bitmap image){
        try {
            Model model = Model.newInstance(getApplicationContext());
```

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

```
        // Creates inputs for reference.
        TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 32, 32, 3}, DataType.FLOAT32);
        ByteBuffer byteBuffer = ByteBuffer.allocateDirect(4 * imageSize * imageSize * 3);
        byteBuffer.order(ByteOrder.nativeOrder());

        int[] intValues = new int[imageSize * imageSize];
        image.getPixels(intValues, 0, image.getWidth(), 0, 0, image.getWidth(), image.getHeight());
        int pixel = 0;
        //iterate over each pixel and extract R, G, and B values. Add those values individually to the byte buffer.
        for(int i = 0; i < imageSize; i ++){
            for(int j = 0; j < imageSize; j++){
                int val = intValues[pixel++]; // RGB
                byteBuffer.putFloat(((val >> 16) & 0xFF) * (1.f / 1));
                byteBuffer.putFloat(((val >> 8) & 0xFF) * (1.f / 1));
                byteBuffer.putFloat((val & 0xFF) * (1.f / 1));
            }
        }

        inputFeature0.loadBuffer(byteBuffer);

        // Runs model inference and gets result.
        Model.Outputs outputs = model.process(inputFeature0);
        TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();

        float[] confidences = outputFeature0.getFloatArray();
        // find the index of the class with the biggest confidence.
        int maxPos = 0;
        float maxConfidence = 0;
        for (int i = 0; i < confidences.length; i++) {
            if (confidences[i] > maxConfidence) {
                maxConfidence = confidences[i];
                maxPos = i;
            }
        }
        String[] classes = {"Apple", "Banana", "Orange"};
        result.setText(classes[maxPos]);

        // Releases model resources if no longer used.
        model.close();
    } catch (IOException e) {
        // TODO Handle the exception
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if(resultCode == RESULT_OK){
        if(requestCode == 3){
            Bitmap image = (Bitmap) data.getExtras().get("data");
            int dimension = Math.min(image.getWidth(), image.getHeight());
            image = ThumbnailUtils.extractThumbnail(image, dimension, dimension);
            imageView.setImageBitmap(image);

            image = Bitmap.createScaledBitmap(image, imageSize, imageSize, false);
```

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

```
                classifyImage(image);
            }else{
                Uri dat = data.getData();
                Bitmap image = null;
                try {
                    image = MediaStore.Images.Media.getBitmap(this.getContentResolver(), dat);
                } catch (IOException e) {
                    e.printStackTrace();
                }
                imageView.setImageBitmap(image);

                image = Bitmap.createScaledBitmap(image, imageSize, imageSize, false);
                classifyImage(image);
            }
        }
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

V. SIMULATION RESULTS

In this research finally we are comparing the dataset with accuracy and loss of the various algorithms. With CNN and Keras we get almost accurate image classification. We get the accurate image classification only when the system is trained well. The accuracy of the system will increase when the system is trained by more number of epochs.

```
Epoch 1/10
23/23 [==============================] - 12s 62ms/step - loss: 1.0528 - accuracy: 0.5261 - val_loss: 0.9706 - val_accuracy: 0.7273
Epoch 2/10
23/23 [==============================] - 2s 52ms/step - loss: 0.7318 - accuracy: 0.7826 - val_loss: 0.7129 - val_accuracy: 0.6970
Epoch 3/10
23/23 [==============================] - 2s 52ms/step - loss: 0.4633 - accuracy: 0.8174 - val_loss: 0.3622 - val_accuracy: 0.8788
Epoch 4/10
23/23 [==============================] - 2s 54ms/step - loss: 0.4245 - accuracy: 0.8239 - val_loss: 0.4387 - val_accuracy: 0.7879
Epoch 5/10
23/23 [==============================] - 2s 52ms/step - loss: 0.2909 - accuracy: 0.8848 - val_loss: 0.2531 - val_accuracy: 0.9242
Epoch 6/10
23/23 [==============================] - 2s 52ms/step - loss: 0.2158 - accuracy: 0.9217 - val_loss: 0.3359 - val_accuracy: 0.9091
Epoch 7/10
23/23 [==============================] - 2s 53ms/step - loss: 0.1935 - accuracy: 0.9217 - val_loss: 0.2496 - val_accuracy: 0.9091
Epoch 8/10
23/23 [==============================] - 2s 53ms/step - loss: 0.1741 - accuracy: 0.9370 - val_loss: 0.2002 - val_accuracy: 0.9091
Epoch 9/10
23/23 [==============================] - 2s 52ms/step - loss: 0.1199 - accuracy: 0.9630 - val_loss: 0.2937 - val_accuracy: 0.9091
Epoch 10/10
23/23 [==============================] - 2s 55ms/step - loss: 0.1028 - accuracy: 0.9609 - val_loss: 0.2875 - val_accuracy: 0.9091
<keras.callbacks.History at 0x7fa184896090>
```

Fig.2. Accuracy and loss for training and testing

In the above image the loss of the system is decreasing for every epoch and the accuracy is increasing for every epoch. This accuracy is evaluated for testing and training modules of the system.

In this module the accuracy and loss of the system is almost the same as the training and testing process.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

```
model.evaluate(test_ds)
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 107ms/step - accuracy: 0.9231 - loss: 0.2781
[0.29287955164909363, 0.9230769276618958]
```

Fig.3.Calculating model accuracy

Visualize training results: Create plots of the loss and accuracy on the training and validation sets.

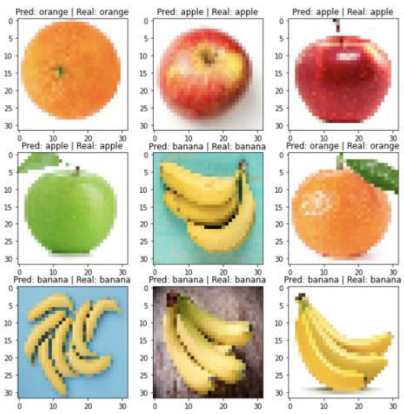


Fig.4. Visualize training and validation results.

V. CONCLUSION AND FUTURE WORK

In conclusion, this research is about image classification by using deep learning via framework TensorFlow. It has three objectives that have achieved throughout this research. The objectives are linked directly with conclusions because it can determine whether all objectives are successfully achieved or not. It can be concluded that all results that have been obtained, showed quite impressive outcomes. The Convolutional neural network (CNN) becomes the main agenda for this research, especially in image classification technology. CNN technique was studied in more details starting from assembling, training model and to classify fruits into categories. The roles of epochs in CNN was able to control accuracy and also prevent any problems such as overfitting. Implementation of deep learning by using framework TensorFlow also gave good results as it is able to simulate, train and classified with up to 96% percent of accuracy towards different types of fruits that have become a trained model. Lastly, Python have been used as the programming language throughout this research since it comes together with framework TensorFlow which leads to designing of the system involved Python from start until ends.

In the future, the image classification using CNNs and Tensorflow involves advancements in architecture, optimization, explainability, adaptability and deployment strategies. By addressing these areas, future work can create more efficient, User-centric and responsible AI solutions suitable for a wide range of industries. To keep pace with the growing demand for efficient and accurate image classification solutions, several future enhancements can be made when using CNNs with the Tensorflow framework. These enhancements focus on improving accuracy, performance, scalability, and user experience.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

1. Mohd Azlan Abu1 , Nurul Hazirah Indra1 , Abdul Halim Abd Rahman1 , Nor Amalia Sapiee1 and Izanoordina Ahmad1,' A study on Image Classification based on Deep Learning and Tensorflow', ***International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 4 (2019), pp. 563-569***
2. Manish U, Karthik Iyer, Md Farhan khan, Nehtravathi.B, Dayananda P,' Image Classification Based On Convolutional Neural Networks Using Tensorflow', ***International Journal of Advanced Research and Publications ISSN: 2456-9992***
3. Mr.Pratik Jadhav, 2Mr.Ashish Kambale, 3Mr.Sairaj Matkar,' IMAGE CLASSIFICATION ANDROID APPLICATION', ***International Journal of Creative Research Thoughts ISSN: 2320-2882***
4. Young Jong Mo, Joongheon Kim, Jong-Kook Kim, Aziz Mohaisen, and Lee,Performance of Deep Learning Computation with TensorFlow Software Library in GPU-Capable Multi-Core Computing Platforms.
5. https://github.com/aparande/Fruit-Classification
6. https://www.tensorflow.org/tutorials/images/classification
7. https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/m.e-cse-batchno-2.pdf
8. https://github.com/ishaanjav