International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN (P): 2249–6831; ISSN (E): 2249–7943
Vol. 11, Issue 1, Jun 2021, 119-128

CNN MODEL FOR IMAGE CLASSIFICATION USING RESNET

Dr. SENBAGAVALLI M[1] & SWETHA SHEKARAPPA G[2*]

[1]*Department of Information Technology, Alliance University, Bangalore, India*

[2] *Department of Electrical and Electronics Engineering, Alliance University, Bangalore, India*

ABSTRACT

Deep artificial neural networks have become a popular study topic in the machine learning and pattern detection communities in current history. As living beings, we tend to categorize things, something that will just come back are often categorified into a class or classes. with in the business, it is a routine troublesome like, categorizing into components, assemblies, fixtures, and merchandise which is a component of the routine. It is often the rationale why living beings started off with algorithms like Machine Learning (ML), Neural Networks (NN), and Deep Learning (DL), among different techniques to alter the method of categorization. In the proposed work we try to implement architectures, ResNet using Keras-Tensorflow library. The efficiency statistics are displayed and debated. ResNet is trained over a large Dataset – Imagenet and from there brings the generalisation aspect of it. There are several trained layers which we can use in our classification. The recommended work's primary focus is to demonstrate methods to construct a CNN model for picture recognition and classification. A customized CNN is imposed and contrasted to a ResNet CNN for the purposes of this research.

KEYWORDS: Convolutional Neural Networks (CNN), Machine Learning (ML), Neural Networks (NN), & Deep Learning (DL)

Original Article

Received: May 05, 2021; **Accepted**: May 25, 2021; **Published**: Jun 01, 2021; **Paper Id.**: IJCSEITRJUN202112

1. INTRODUCTION

Image perception is also known as computer vision and it refers to technologies that identify the objects and several other variables in digital images. The main advantages of using convolutional layers over other fully connected layers are as follows, first the parameter sharing and the next is the sparsity of connections. Edge recognition is a common task that is essential for a variety of traditional image processing processes (such as picture authentication and classification) as well as recent tasks like picture-to-image conversion, photo sketching, and many more. The main objective of Deep Convolutional model is to understand and implement a residual network and to analyse the dimensionality reduction of a volume in a very deep network. The major focus of the work is purely image classification using CNN. The "Fruits 360 Database" will be used because the goal of this component is to analyze the classification method. The following are the major parameters of this database: actual no. of pictures:90583, training set size:67682, pictures (1 fruit or vegetable per picture), test set size: 22788 pictures (1fruit or vegetable per picture), no. of classes: 131 (vegetables and fruits), picture size-100x100 pixels.

Get Training and Testing Data

Ingesting the testing and training datasets is done in Keras by a handy document assistant. Create a “training” & “validation” folder and then place the required pixel. When using a multi-category predictor, such as which is used in this work, every image category has its specific folder —” Apple”, “Watermelon”, etc. flow_from_directory() will undoubtedly adjust the design from the folder design of the directory pertaining pictures. Every subsection of the strength and conditioning (or verification) would be an objective classification.

Figure 1.1: Source Image – Avocado.

Despite the fact that we can readily tell that it is an avocado, the NN will only see the outcome of certain feature extraction filtering. As a result, an avocado will appear as a greenish spherical object in the picture's center. You may get a mediocre forecast if you retrain the NN with precisely centred avocados and afterwards give it an off-centre avocado.

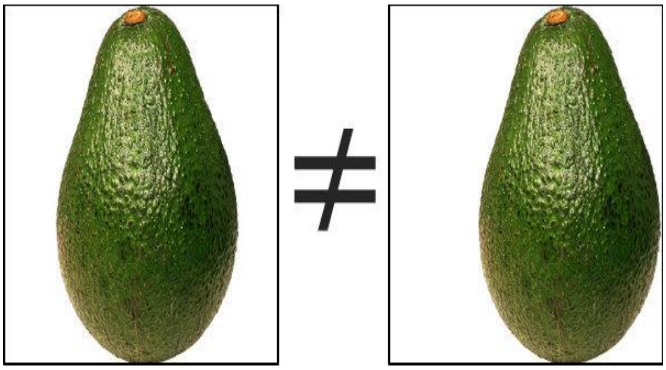

Figure 1.2: Object Comparison.

Keras provides the Image Generator software to handle this and make the instructional program easier. It helps specify the setup parameters by enriching the dataset and implementing arbitrary modifications in luminance, spin, magnification, and distorting. This is a method of arbitrarily expanding the set of data and increasing the instructional data's resilience.

2. LITERATURE SURVEY

In [1] the authors proposed a new database called “ImageNet”, this provides a full examination of Image classification in its existing condition which includes 12 substrings, 5247 synsets, and 3.2 million photos overall. The data from the Image Net Large Scale Visual Recognition Challenge (ILSVRC) was used in a huge investigation for Detecting avocados to Zucchinis was proposed in [2]. In [3,] the authors aim to integrate and organize current transfer acquisition scientific studies, and also contextualise the mechanics and tactics of transferable learning in a coherent manner. The paper [4] describes a strategy for recognizing fruit greater quickly and precisely by using the transfer learning technique. An experimental study of prominent convolutional neural networks' (CNNs) ability in detecting entities in actual live streams by authors in [5] was done in which Google Net and ResNet50,when contrasted to Alex Net, can distinguish artifacts with more resolution. The recommended method creates narrow edge-maps that look natural to the naked eye by authors in [6] which is energized on both HED (Holistically-Nested Edge Detection) and Xception networks. In this paper [7], it describes the TensorFlow dataflow paradigm and showcases TensorFlow's impressive effectiveness in a variety of real-world scenarios. In this paper [8] the author provides newly enacted creative approaches but disregards addressing the fundamentals, allowing readers to understand the program's latest technology more effortlessly. Furthermore, unlike earlier image retrieval studies, in [9] the authors conducted a thorough and extensive evaluation of deep learning object identification approaches, as well as the most up-to-date identification solutions and a collection of notable emerging technologies. In [11] the authors have proposed an Architecture, which the Zynq-7000 AP SoC Processing System (PS) and Programmable Logic(PL) are the two primary operational pieces. In [12] the authors proposed a hybrid approach for feature selection which is the main part of any type of Detection mechanism. Above all, we want to give you an understanding of how various deep learning approaches are applied and based on it that the proposed work is shown with the way to create a CNN model for picture classification and identification.

3. EXISTING SYSTEM

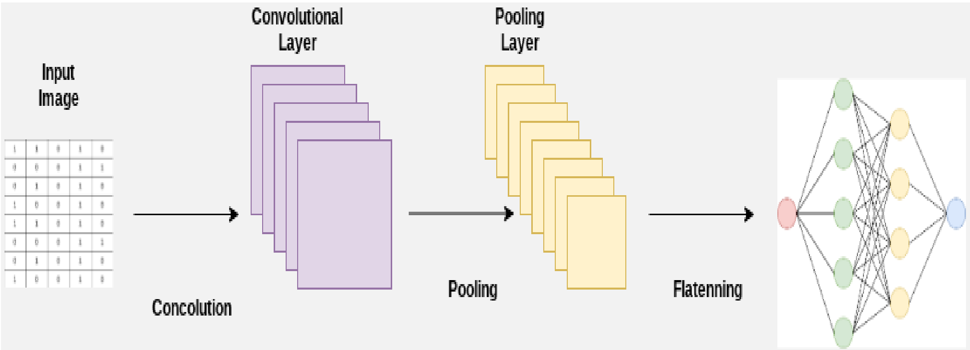

Figure 3.1: "The Convolutional Process".

So, what is a convolutional neural network, and how does it operate? A convolutional neural network is simply an additional way to create a network (model) to provide correct categorization. Convolutional neural networks outperform most other neural networks when it comes to computerized image processing. What renders CNNs or (covnets) so great is that the structures they acquire from visuals are translate irreversible which means that if they detect a motif in one image's sector, they will perceive it in any other object's edge, but a standard system will have to re-learn it. Covnets could also acquire spatiotemporal data topologies, which implies that each stratum of a Covnet will acquire something distinctive. The primary level may acquire little motifs, whereas the second level may acquire larger correlations that are linked to the first gradient attributes. The convolutional algorithm is used to attain these characteristics.

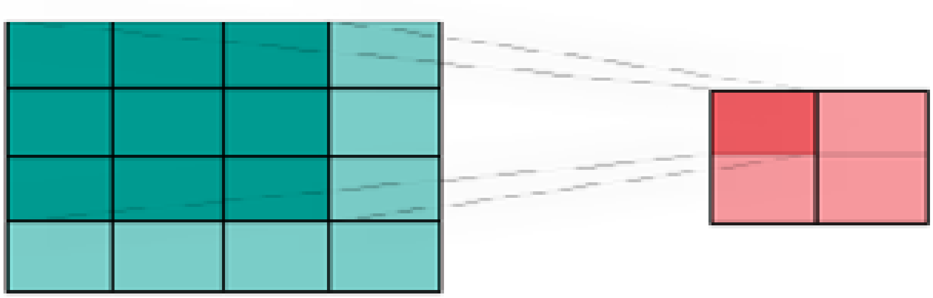

Figure 3.2: "A Gif of the Movement of a Convolution".

A. Convolution

A convolutional layer's characteristic mappings (character vectors) are obtained using the convolution function. covnets have set loads that are made up of a kernel by default. A kernel is utilized to extract distinguishing properties from source images (input layer), such as crispness and margins, or to collect data on how to recognize a boundary. This algorithm can be written as n*n, that is a grid with a large number of distinct elements. Let's say the activation function is(10,10) as well as the kernel is, (3,3).The initial phase (stride) will multiply the source picture by 9 pixels in the exact upper left quadrant to build a particular image in the upper left quadrant of synthetic matrices termed a feature map.As the kernel moves throughout the source images, the doubling proceeds.

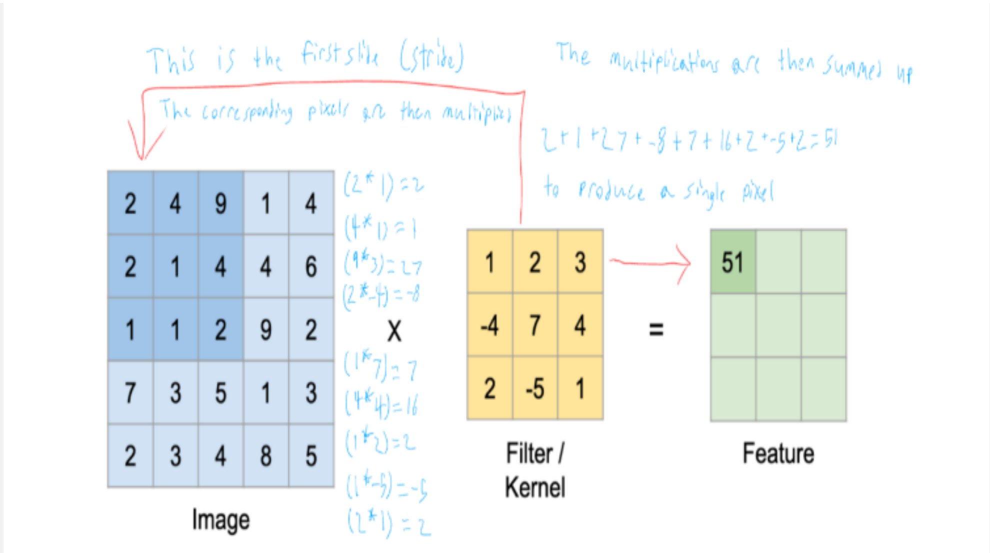

Figure 3.3: Krut Patel from Towards Data Science, 2019.

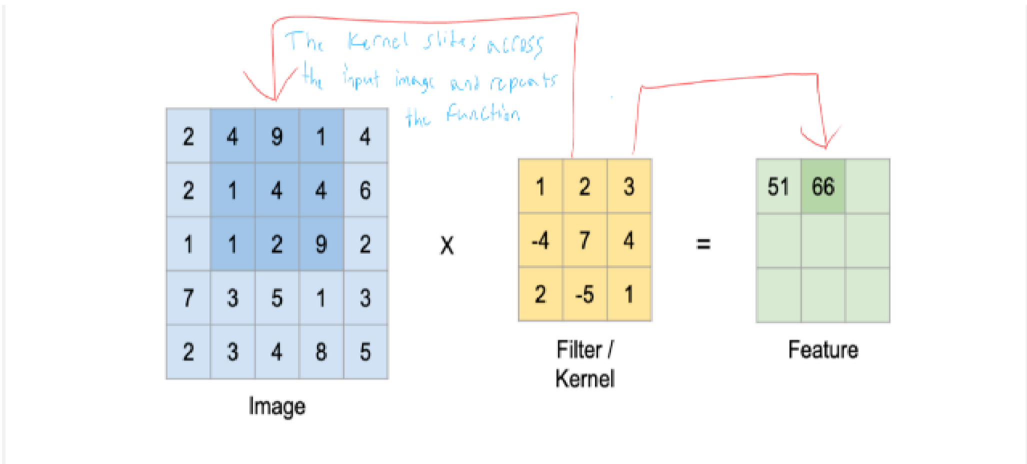

Figure 3.4: Krut Patel from Towards Data Science, 2019.

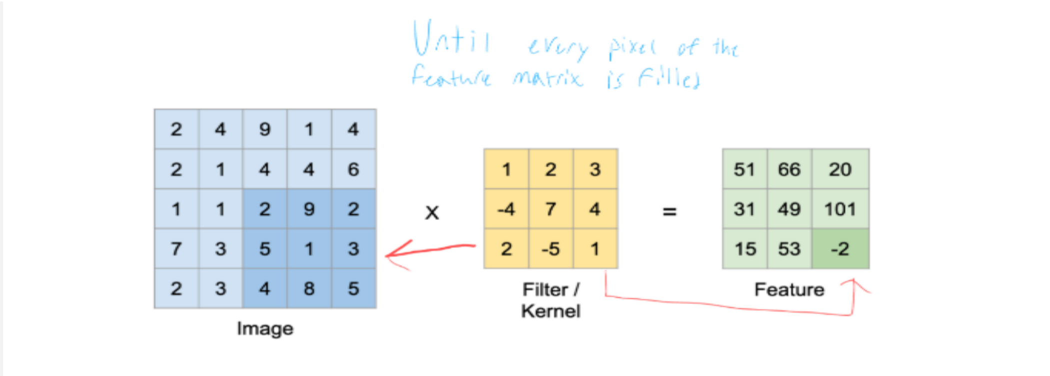

Figure 3.5: Krut Patel from Towards Data Science, 2019.

This continuous operation continues until all of the trait matrix's complicated values have been filled. The enhanced image is piled within a convolution operation while it is finished. If the model is programmed to do so, some other kernel will generate a new enhanced image from the identical source images, which will be stored in the similar convolution layers as the previous feature vector.

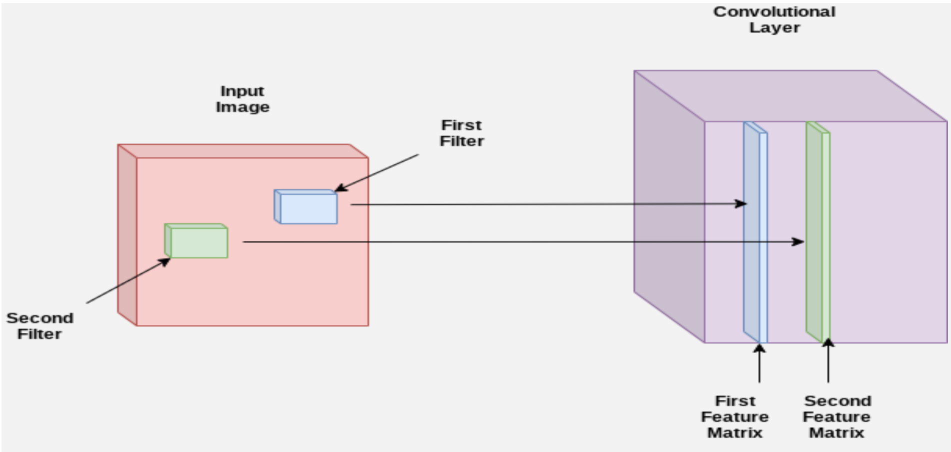

Figure 3.6: "What Happens when Feature Matrices are Made".

4. PROPOSED SYSTEM

Now it is time to show how transferable learning's abilities can be put to use in the real world. We'll go over the information being used. The chosen well before modeling, the model topology, and finally the code. Deep learning works on the fundamental assumption of adopting a pretrained model on a huge database and transferring its information to a lower dimension. Isolate the initial convolution layers of cells and just retrain the last several levels that provide a forecast for image processing with a Convolutional Neural Network. The theory is that the convolutional layers capture broad, minimal characteristics like boundaries, motifs, and gradient that really are relevant throughout frames, while the subsequent levels detect key features inside an object like eyeballs or axles.As there are ubiquitous, close to zero properties conserved by photos, we would leverage a system trained on irrelevant classes in a massive dataset (typically Imagenet) and adapt it towards our own challenge.

4.1 Proposed System Architecture

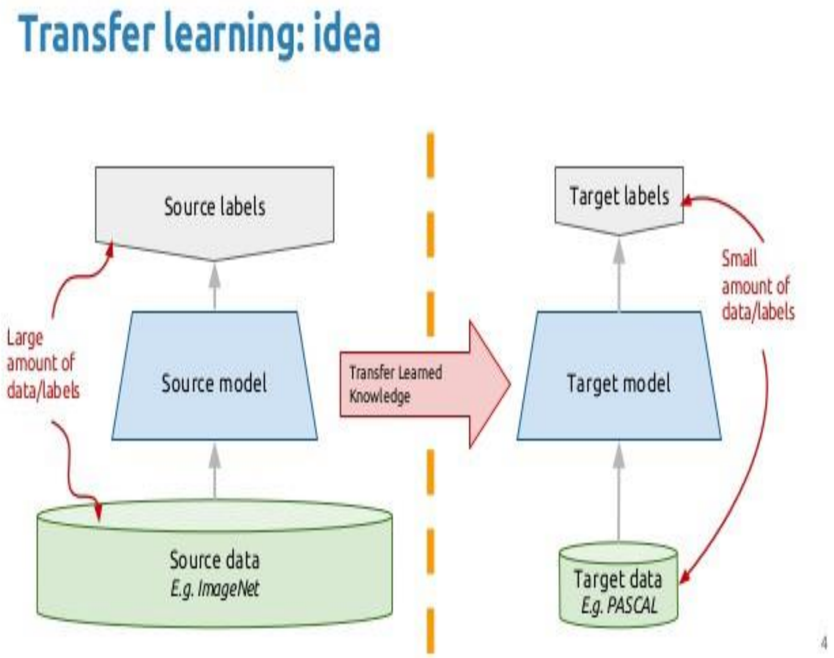

Figure 4.1: Transfer Learning

The below is a broad framework for image detection transfer learning:

- Load information into a post CNN model that has been developed on a big database.
- Preserve the figure's lowest convolution levels' variables (weights).
- To the system, build an unique filter with numerous levels of employable attributes
- Use the given learning algorithm to learn the classifying levels.
- If necessary, perfect model parameters and trans theoretical further levels.

Plain CNN Execution on the Fruits-360 dataset for a few categories:

```
deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/20
89/89 [==============================] - 45s 500ms/step - loss: 2.6692 - accuracy: 0.3709 - val_loss: 2.5591 - val_accuracy: 0.1541
Epoch 2/20
89/89 [==============================] - 43s 481ms/step - loss: 0.8780 - accuracy: 0.7089 - val_loss: 3.0915 - val_accuracy: 0.2159
Epoch 3/20
89/89 [==============================] - 43s 484ms/step - loss: 0.6341 - accuracy: 0.7965 - val_loss: 2.8869 - val_accuracy: 0.2981
Epoch 4/20
89/89 [==============================] - 42s 479ms/step - loss: 0.5367 - accuracy: 0.8289 - val_loss: 1.9866 - val_accuracy: 0.4801
Epoch 5/20
89/89 [==============================] - 43s 488ms/step - loss: 0.3965 - accuracy: 0.8692 - val_loss: 0.8584 - val_accuracy: 0.7489
Epoch 6/20
89/89 [==============================] - 43s 488ms/step - loss: 0.3248 - accuracy: 0.8890 - val_loss: 0.6013 - val_accuracy: 0.8442
Epoch 7/20
89/89 [==============================] - 49s 556ms/step - loss: 0.2749 - accuracy: 0.9187 - val_loss: 0.5255 - val_accuracy: 0.8731
Epoch 8/20
89/89 [==============================] - 47s 525ms/step - loss: 0.2411 - accuracy: 0.9329 - val_loss: 0.5012 - val_accuracy: 0.8891
Epoch 9/20
89/89 [==============================] - 45s 512ms/step - loss: 0.2306 - accuracy: 0.9242 - val_loss: 0.5063 - val_accuracy: 0.9044
Epoch 10/20
89/89 [==============================] - 49s 548ms/step - loss: 0.1876 - accuracy: 0.9400 - val_loss: 0.5828 - val_accuracy: 0.8961
Epoch 11/20
89/89 [==============================] - 44s 492ms/step - loss: 0.1930 - accuracy: 0.9404 - val_loss: 0.4998 - val_accuracy: 0.9013
Epoch 12/20
89/89 [==============================] - 43s 481ms/step - loss: 0.1729 - accuracy: 0.9482 - val_loss: 0.5648 - val_accuracy: 0.9073
Epoch 13/20
89/89 [==============================] - 45s 503ms/step - loss: 0.1300 - accuracy: 0.9553 - val_loss: 0.5959 - val_accuracy: 0.8881
Epoch 14/20
89/89 [==============================] - 44s 496ms/step - loss: 0.1234 - accuracy: 0.9592 - val_loss: 0.6487 - val_accuracy: 0.9020
```

Transfer Learning based execution of complete dataset of fruits-360, Resnet 50 was used:

```python
# Execute the model with fit_generator within the while loop utilizing the discovered GPU
import tensorflow as tf
with tf.device("/device:GPU:0"):
    history = model_ResNet50.fit_generator(
        train_generator,
        epochs=5,
        validation_data=test_generator,
        verbose = 1,
        callbacks=[EarlyStopping(monitor='val_accuracy', patience = 5, restore_best_weights = True)])
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: `Model.fit_generator` is deprecate
d and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
```

```
Epoch 1/5
4231/4231 [==============================] - 2105s 495ms/step - loss: 2.7168 - accuracy: 0.5058 - val_loss: 0.1295 - val_accuracy: 0.97
25
Epoch 2/5
4231/4231 [==============================] - 1794s 424ms/step - loss: 0.1147 - accuracy: 0.9879 - val_loss: 0.0608 - val_accuracy: 0.98
65
Epoch 3/5
4231/4231 [==============================] - 1835s 434ms/step - loss: 0.0460 - accuracy: 0.9959 - val_loss: 0.0473 - val_accuracy: 0.98
84
Epoch 4/5
4231/4231 [==============================] - 1848s 437ms/step - loss: 0.0276 - accuracy: 0.9978 - val_loss: 0.0380 - val_accuracy: 0.99
02
Epoch 5/5
4231/4231 [==============================] - 1781s 421ms/step - loss: 0.0195 - accuracy: 0.9988 - val_loss: 0.0360 - val_accuracy: 0.99
11
```

5. CONCLUSIONS

The work depicted in this paper is that the goal of a convolutional neural network is to find all the proper data for each of the kernels. In this proposed work we have implemented architectures, ResNet using Keras-Tensorflow library. The major focus of the work is purely image classification using CNN. The reliability findings are displayed and evaluated. As a result, whenever the source visual is processed through the levels, various neurons on the penultimate hidden layers are triggered, allowing the vision to be predicted and classified appropriately.

REFERENCES

1. *Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255. Ieee, 2009.*
2. *Swetha Danthala, Seeramsrinivasa Rao, Kasiprasad Mannepalli & Dhantala Shilpa, "Robotic Manipulator Control by Using Machine Learning Algorithms: A Review", International Journal of Mechanical and Production Engineering Research and Development (IJMPERD),Vol. 8, Issue 5,pp, 305-310*
3. *Russakovsky, Olga, Jia Deng, Zhiheng Huang, Alexander C. Berg, and Li Fei-Fei. "Detecting avocados to zucchinis: what have we done, and where are we going?." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2064-2071. 2013.*
4. *B. Satish Kumar & Y. Kalyan Chakravarthy, "Prediction of Optimal Torques from Gait Analysis Applying the Machine Learning Concepts", International Journal of Mechanical and Production Engineering Research and Development (IJMPERD),Vol. 9, Issue 4, pp, 685-698*
5. *Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. "A comprehensive survey on transfer learning." Proceedings of the IEEE 109, no. 1 (2020): 43-76.*
6. *N. S. Chandolikar & V. D. Nandavadekar, "Investigation of Feature Selection and Ensemble Methods for Performance Improvement of Intrusion Attack Classification ",International Journal of Computer Science and Engineering (IJCSE)Vol. 2, Issue 3, pp, 131-136*

7. *Huang, Ziliang, Yan Cao, and Tianbao Wang. "Transfer Learning with Efficient Convolutional Neural Networks for Fruit Recognition." In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 358-362. IEEE, 2019.*

8. *G. Kumaresan & P. Rajakumar, "Predictive Analytics Using Big Data: A Survey ",BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE),Vol. 3, Issue 9,pp, 61-68*

9. *Sharma, Neha, Vibhor Jain, and Anju Mishra. "An analysis of convolutional neural networks for image classification." Procedia computer science 132 (2018): 377-384.*

10. *Poma, Xavier Soria, Edgar Riba, and Angel Sappa. "Dense extreme inception network: Towards a robust cnn model for edge detection." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1923-1932. 2020.*

11. *Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "Tensorflow: A system for large-scale machine learning." In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp. 265-283. 2016.*

12. *Zhang, Yulun, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. "Residual dense network for image super-resolution." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2472-2481. 2018.*

13. *Silberman, Nathan, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. "Indoor segmentation and support inference from rgbd images." In European conference on computer vision, pp. 746-760. Springer, Berlin, Heidelberg, 2012.*

14. *Mély, David A., Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. "A systematic comparison between visual cues for boundary detection." Vision research 120 (2016): 93-107.*

15. *Eetha, Sagar, Sonali Agrawal, and Srikanth Neelam. "Zynq FPGA based system design for video surveillance with Sobel Edge Detection." In 2018 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), pp. 76-79. IEEE, 2018.*

16. *Valli, M. S., and G. T. Arasu. "An Efficient Feature Selection Technique of Unsupervised Learning Approach for Analyzing Web Opinions." (2016).*

17. *Senbagavalli, M 2018, 'A Combined Approach for Supervised Learning of Online User Reviews using Opinion Mining', International Journal of Research Review In Engineering And Management, Volume-2, Issue 2,P.NO.1-21.*