IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 9, Issue 11, November 2020

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.512

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

Insect Classification using Custom CNN Vs Transfer Learning

Naumanurrahman Shaikh Mujiburrahman[1], Manish Sanjay Zalte[2], Dr. Manoj E. Patil[3]

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India[1]

UG Student, Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India[2]

Assoc. Prof., Dept. of Computer Engineering, SSBT COET Bambhori, Jalgaon, India [3]

ABSTRACT: For Insect Classification there are many methods proposed. To find the more suitable classifier we have implemented two different methods of classification, Custom CNN and Transfer Learning. We observed the accuracy and loss parameters during training phase and validation phase on both Custom CNN and Transfer learning methods. In Transfer Learning we have created the base model from the pre-trained model MobileNetV2. This model is further trained on Imagenet Dataset which consists of 1.2M labeled images .We compared all attributes of both models on this dataset. Recorded Accuracy and Loss for training and Validation. From the results we found that Custom CNN model works better than Transfer Learning for this dataset. This model can also be used for education and farming purpose.

KEYWORDS: CNN, Transfer Learning, MobileNetV2, Accuracy, Loss , Training, Validation, Imagenet, Pooling Dense

I. INTRODUCTION

[4]Insects are the most diverse group of animals, they include more than a million described species and represent more than half of all known living organisms. The total number of extant species is estimated at between six and ten million. Insects may be found in nearly all environments, although only a small number of species reside in the oceans. Insects can be more interest in many fields such as Agriculture, Industrial, Research and Studies.

Humans classify insects as pests where certain insects can play major role in agriculture economy by either helping the production of crops or infecting the crop depending on the need. Industrial use of insects are the products that we use in our day-to-day life products such as silk and honey are most common example industrilation .

These are all the Researched based example, where people with prior knowledge study insects. But there are lot of people who are curious to know about insects and there growth period and do not have knowledge about insects. This model can help those people to identify different type of insects. Usually this is done by Agricultural expertise manually Although it is not possible to classify and identify different types of insect manually for person without prior knowledge about insects.To overcome this problem it is necessary to develop a way that is effective and rapidly.

In this paper we have worked on the method of insect classification using Custom CNN and Transfer Learning and discussed which method will be best for insect classification using this dataset. We have judged both models on the basis of Accuracy and Loss. We used log loss as our metric for calculating the loss. The plot of Accuracy and Loss are plotted using matplotlib and and are shown which shows the performance of models.

A. Convolutional

When programming a CNN, the input is a tensor with shape (number of images) x (image width) x (image height) x (image depth). Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map width) x (feature map height) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- Convolutional kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameter).
- The depth of the Convolution filter (the input channels) must be equal to the number channels (depth) of the input feature map.

B. Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Average pooling uses the average value from each of a cluster of neurons at the prior layer.

C. Fully Connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

D. Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning, in a neural network, progresses by making iterative adjustments to these biases and weights. The vector of weights and the bias are called filters and represent particular features of the input (e.g., a particular shape).

E. Activation Functions

Activation layers apply a non-linear operation to the output of the other layers such as convolutional layers or dense layers.

• ReLu Activation: ReLu or Rectified Linear Unit computes
the function $f(x)=max(0,x) to threshold the activation at 0.

• SoftMax Activation: SoftMax function is applied to the output layer to convert the scores into probabilities that sum to 1.

F. Optimizer

• Adam (Adaptive moment estimation): is an update to RMSProp optimizer in which the running average of both the gradients and their magnitude is used. In practice Adam is currently recommended as the default algorithm to use, and often works slightly better than RMSProp. In my experiments Adam also shows general high accuracy while Adadelta learns too fast. I've used Adam in all the experiments because I felt having similar optimizer would be a better baseline for comparing the experiments.

G. ImageNet

The ImageNet project is inspired by a growing sentiment in the image and vision research field – the need for more data. Ever since the birth of the digital era and the availability of web-scale data exchanges, researchers in these fields have been working hard to design more and more sophisticated algorithms to index, retrieve, organize and annotate multimedia data. But good research needs good resource. To tackle these problem in large-scale (think of your growing personal collection of digital images, or videos, or a commercial web search engine's database), it would be tremendously helpful to researchers if there exists a large-scale image database. This is the motivation for us to put together ImageNet. [2]

H. MobileNetV2

MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depth wise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.[3]

II. METHODOLOGY

A. Custom CNN

A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analysing visual imagery. CNNs are regularized versions of multilayer perceptron (fully connected networks).The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

convolution in place of general matrix multiplication in at least one of their layers. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. The final convolution, in turn, often involves backpropagation in order to more accurately weight the end product.

Network Architecture

A 9-layered network is used to predict the class of each Insect. The following layers are used in the ConvNet.
Found 63364 images belonging to 291 classes.
Found 21207 images belonging to 291 classes.
Model: "sequential"

Layer (type)	Output Shape	Parameter #
conv2d (Conv2D)	(None,254,254, 16)	448
max_pooling2d (MaxPooling2D)	(None,127,127, 16)	0
conv2d_1 (Conv2D)	(None,125,125, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 64)	0
flatten (Flatten)	(None, 57600)	0
dense (Dense)	(None, 512)	29491712
dense_1 (Dense)	(None, 291)	149283

Total parameters: 29,664,579
Trainable parameter: 29,664,579
Non-trainable parameter: 0

Layers:
• Convolution: Convolutional layers convolve around the image to detect edges, lines, blobs of colors and other visual elements. Convolutional layers hyper parameters are the number of filters, filter size, stride, padding and activation functions for introducing non-linearity.
• MaxPooling: Pooling layers reduces the dimensionality of the images by removing some of the pixels from the image. MaxPooling replaces a n x n area of an image with the maximum pixel value from that area to down sample the image.
• Flatten: Flattens the output of the convolution layers to feed into the Dense layers.
• Dense: Dense layers are the traditional fully connected networks that maps the scores of the Convolutional.

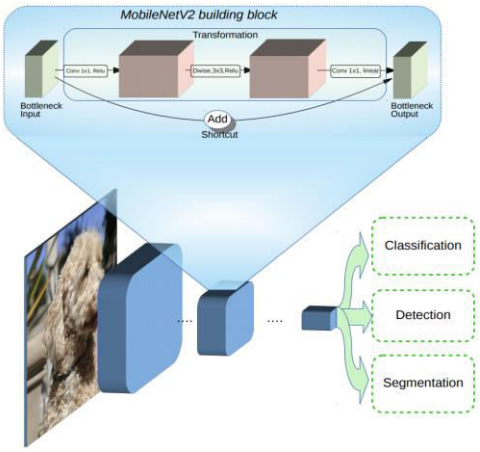
B. Transfer Learning

Transfer Learning: Transfer learning refers to the process of using the weights of a pre-trained network trained on a large dataset applied to a different dataset (either as a feature extractor or by fine-tuning the network).Fine-tuning refers to the process of training the last few or more layers of the pre-trained network on the new dataset to adjust the weight. Transfer learning is very popular in practice as collecting data is often costly and training a large network is computationally expensive. Here, in this case we have used weights from a convolutional neural network pre-trained on ImageNet dataset is fine-tuned to classify weather condition.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

Network Architecture



In this method we are using the pre-trained model of MobileNetV2
Model: "mobilenetv2_1.00_224"
Total parameters: 2,257,984
Trainable parameters: 0
Non-trainable parameters: 2,257,984
Found 63364 images belonging to 291 classes.
Found 21207 images belonging to 291 classes.
Model: "sequential"

Layer (type)	Output Shape	Parameter #
mobilenetv2_1.00_224	(Functi (None, 8, 8, 1280)	2257984
global_average_pooling2d	(Gl (None, 1280)	0
dense (Dense)	(None, 291)	372771

Total parameters: 2,630,755
Trainable parameters: 372,771
Non-trainable parameters: 2,257,984

C. Metric

Here, I'm using the metric for this project is sparse categorical cross entropy

$$logloss = \frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log(p_{ij})$$

Here, each image has been labelled with one true class and for each image a set of predicted probabilities should be submitted. N is the number of images in the test set, M is the number of image class labels, log is the natural logarithm, yij is 1 if observation i belongs to class j and 0 otherwise, and pij is the predicted probability that observation i belongs to class j.
A perfect classifier will have the log-loss of 0.
Multiclass log-loss punishes the classifiers which are confident about an incorrect prediction. In the above equation, if the class label is 1(the instance is from that class) and the predicted probability is near to 1(classifier predictions are correct), then the loss is really low as
$\log(x) \rightarrow 0$ as $x \rightarrow 1$, so this instance contributes a small amount of loss to the total loss and if this occurs for every single instance(the classifiers is accurate) then the total loss will also approach 0.
On the other hand, if the class label is 1(the instance is from that class) and the predicted probability is close to 0(the classifier is confident in its mistake), as log(0) is undefined it approaches $\rightarrow\infty$ so theoretically the loss can approach

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

infinity. In order to avoid the extremes of the log function, predicted probabilities are replaced with max(min(p, 1-1015),1015) .
Graphically[^1] , assuming the i^{th} instance belongs to class j and yij = 1 , it's shown that when the predicted probability approaches 0, loss can be very large.

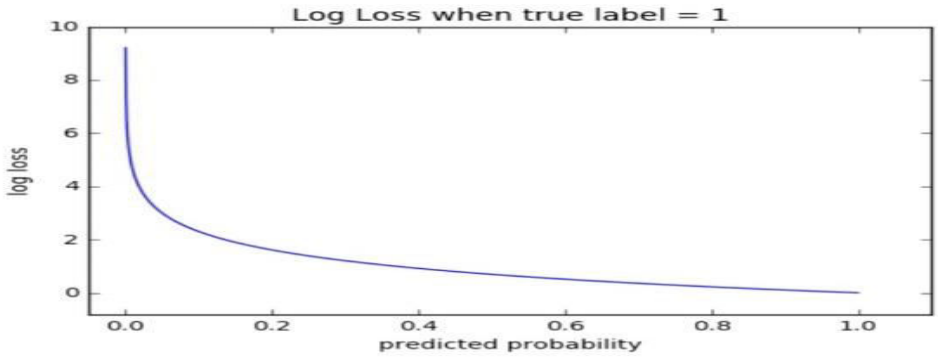


Fig:1.

III. IMPLEMENTATION

A. Dataset

This dataset contain 63364 specimens that we used to train, validate and test a convolutional neural network. This Dataset is created in 2017 which now contains 291 types of insects[1]. In this dataset each folder is named as the gbif id number. E.g. Carabus problematicus is 4470555.
We changed those gbif id numbers to the real species name while data preprocessing and created a dataset with 291 classes named as species name. We have also splited dataset into training and validation set where in training set 63364 images belonging to 291 classes. And in validation set 21207 images belonging to 291 classes.

B. Custom CNN

While training the custom CNN we use Image Data Generator to ease Data preparation as it lables images based on Folder Name which is ideal for the way Data Set is arranged. In the process of Image Data Generation we provided directory of training and validation data set, in the image data generation class_mode is set as “binary”, batch_size is equal to 100 and the image shape is defined as 256x256.
For model purpose we used Tensorflows Keras Sequential model with 9 layers.

Layer1: Input layer Conv2D with activation function RELU with input shape = (256, 256, 3).
Layer2: It is a pooling layer which is MaxPooling2D(2, 2).
Layer3:A convolutional layer Conv2D with activation function RELU.
Layer4: It is also a pooling layer which is MaxPooling2D(2,2).
Layer5: It is also a convolutional layer Conv2D with activation function RELU.
Layer6: It is also a pooling layer which is MaxPooling2D(2,2).
Layer7:It is a Flatten Layer.
Layer8:It is a Dense layer with activation function RELU.
Layer9:It is a Dense layer with activation function softmax.

The Model is trained over 63364 images belonging to 291 classes. And Validate over 21207 images belonging to 291 classes. Total number of parameters are 29,664,579.Adam Optimizer is used in the training of the model as metric we use spares_categorical_crossentropy.The model is trained over 10 epoch . We retrieve a list of results on training and test data sets for each training epoch which is used to plot training and validation accuracy per epoch.
After 10Epoch model is saved as CNNmodel.h5.

C. Transfer Learning

While training model using transfer learning we used Keras Image Data Generator to generate image data with size defined as (256, 256, 3) .We have created the base model from the pre-trained model MobileNet V2 which is trained on imagenet Dataset which consists of 1.2M labelled images, where size equal to the defined size ,weights set to

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

"imagenet" and include_top="False". To tell Tensorflow not to adjust weigths of imagenet model which are already trained we set"imagenet.trainable = False". We defined the new model with imagenet as the Base Model using Sequential Api. with two more added layers.

Layer1:GlobalAveragePooling2D.
Layer2:Dense layer with activation function softmax.

We used Image Data Generator to ease Data preparation as it lables images based on Folder Name which is ideal for the way Data Set is arranged. In the process of Image Data Generation we provided directory of training and validation data set, in the image data generation class_mode is set as "binary", batch_size is equal to 100 and the image shape is defined as 256x256. Using Adam as it works well with image classification and can adjust Learning rate while training unlike Gradient Descent Optimizer where manual LR tuning needs to be done. Using sparse_categorical_crossentropy as Loss function for similar reasons. The model is trained over 10 epoch . We retrieve a list of results on training and test data sets for each training epoch which is used to plot training and validation accuracy per epoch.
After 10 epoch Model is saved as TLmodel.h5

IV. RESULTS AND DISCUSSION

A. Custom CNN

After completion of 10 epochs Accuracy and loss recorded for Training and Validation are as follows.

Plots

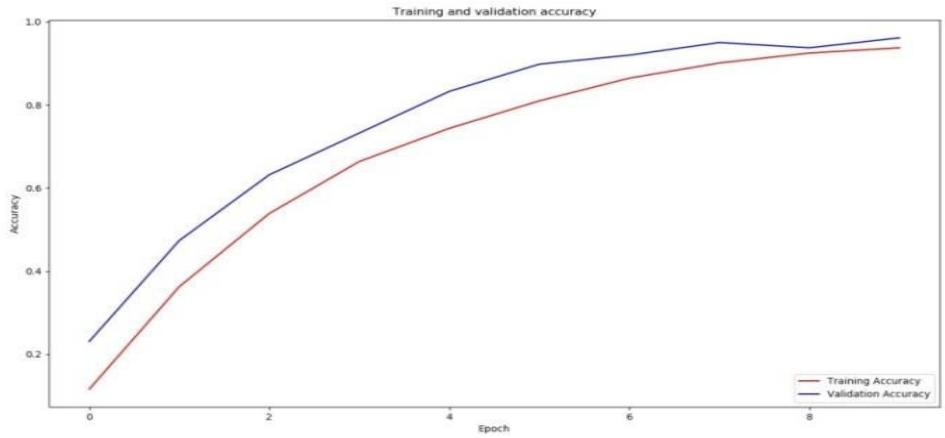


Fig:2 Training And Validation Accuracy

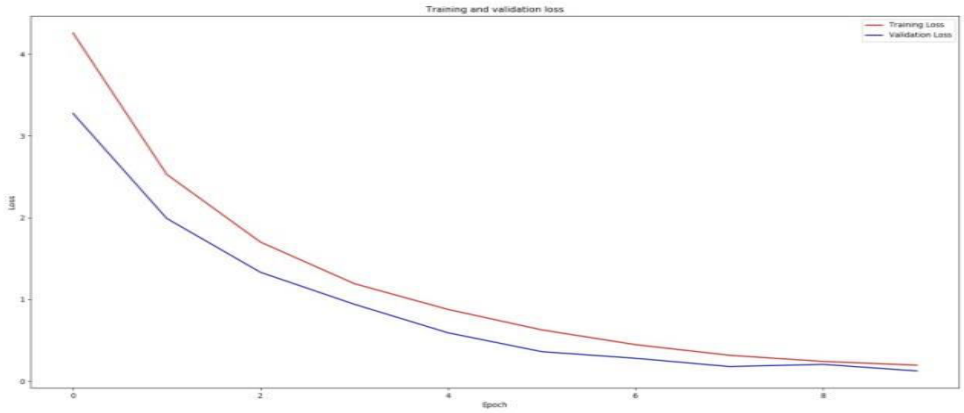


Fig:3 Training And Validation Loss

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

Training:

Accuracy=0.790

Loss=0.301

Validation:

Accuracy=0.860

Loss=0.265

B. Transfer Learning

After completion of 10 epochs Accuracy and loss recorded for Training and Validation are as follows.

Plots

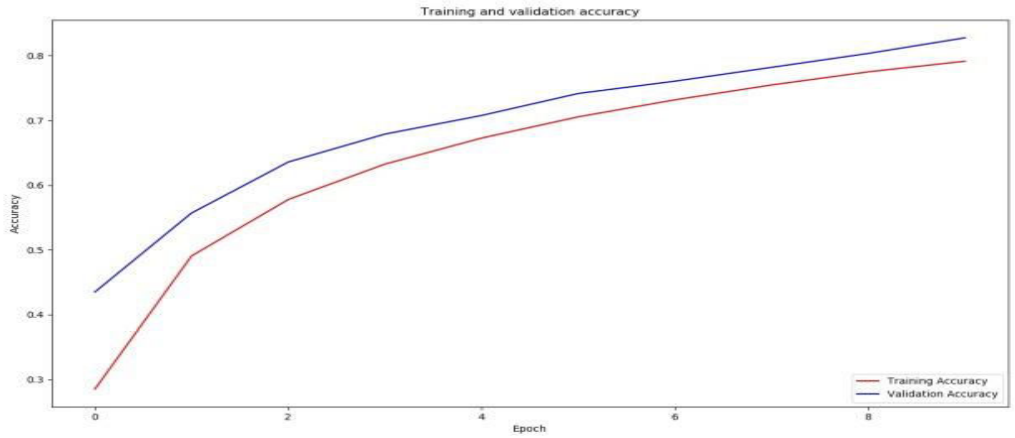


Fig:4 Training And Validation Accuracy

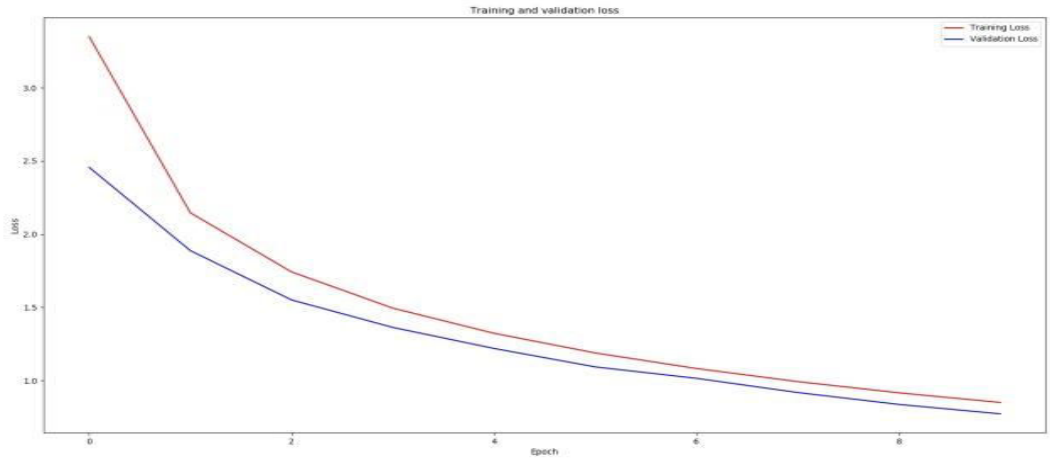


Fig:5 Training And Validation Loss

Training:

Accuracy=0.940

Loss=0.556

Validation:

Accuracy=0.965

Loss=0.458

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 11, November 2020 ||

V. CONCLUSION AND FUTURE SCOPE

A. Conclusion

After implementation of both Costum CNN and Transfer Learning on the dataset of insect classification and recording the accuracy and loss of both training and validation on of both methods we concluded that the accuracy of the Custom CNN is better than Transfer learning in both training and validation. The Loss of Custom CNN is less than Transfer Learning in both validation and training. Thus it is observed that Custom CNN is more suitable for this used dataset than Transfer Learning Due the Big Size of the dataset another reason is the presence 291 classes in the dataset.

B. Future Scope

The model created using CNN can be used in future to develop application for insect detection and classification using live video source input which will we be used by different insect enthusiast and student to learn about the insect species. This method can also have application in farming sector to identify the insect which causing disease.

REFERENCES

[1] Image data used for publication "Species-level image classification with convolutional neural network enable insect identification from habitus images "J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.(https://zenodo.org/record/3549369#.X7IGM2gzZPZ)

[2] Patel, Z., Senjaliya, N., & Tejani, A. (2019). AI-enhanced optimization of heat pump sizing and design for specific applications. International Journal of Mechanical Engineering and Technology (IJMET), 10(11), 447-460.

[3] http://www.image-net.org

[4] V.Sudarsan, R.Sugumar, Building a distributed K-Means model for Weka using remote method invocation, Concurrency and Computation Practice and Experience, Volume 31, Issue 5, July 2019.

[5] https://paperswithcode.com/method/mobilenetv2

[6] https://en.wikipedia.org/wiki/Insect#:~:text=Insects%20have%20a%20chitinous%20exoskeleton,of%20all%20known%20living%20organisms

[7] **B.Murugeshwari,** C Jayakumar, , K Sarukesi, Preservation of the Privacy for Multiple Custodian Systems with Rule Sharing , **Journal of Computer Science**, Vol No 09 Issue 09, 1086-1091 Pages, 2013 **ISSN : 1549-3636**

[8] https://github.com/prashant-raghu/Multiclass-scene-classification-using-tf.keras

[9] B.Vijayalakshmi, R.Sugumar, Rough set theory based feature selection and FGA-NN classifier for medical data classification, International Journal of Business Intelligence and Data Mining, Volume 14, Issue 3, pp.322-358, Feb 2019.

[10] Senjaliya, N., & Tejani, A. (2020). Artificial intelligence-powered autonomous energy management system for hybrid heat pump and solar thermal integration in residential buildings. International Journal of Advanced Research in Engineering and Technology (IJARET), 11(7), 1025-1037.

[11] https://github.com/vijayg15/Keras-MultiClass-Image-Classification

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.512

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details