Satellite View Classification of Terrains

Mr. Abhimanyu Singh (11800740)
Department of Computer Science & Engineering
Lovely Professional University, Phagwara, India
abhimanyusinghchauhan98@gmail.com

Mr. Shaktish Prajapati ()
Department of Computer Science & Engineering
Lovely Professional University, Phagwara, India
pshaktish@gmail.com

Mr. Sagar Pande
Department of Computer Science & Engineering
Lovely Professional University, Phagwara, India
sagar.23754@lpu.co.in

ABSTRACT: -

Image classification is a procedure where the algorithm attempts to extract highlights from the image gave and attempt to anticipate to which classification it has a place with or it very well may be increasingly explicit and go to a stretch out of saying what the picture really is relies upon the manner in which the algorithm is planned the model turns out to be increasingly more accurate when it is furnished with an ever increasing number of data before the introduction of deep learning and neural networks this was a boisterous errand and taking care of this gigantic data and creating a precise model was not practical but rather after the introduction of previously mentioned ideas researchers thought of numerous plans to extract features from the data and to make the task significantly simpler Google concocted tensorflow an open source library for machine learning lovers this paper makes reference to how deep learning can be utilized to make predictions and use it to build an image classifier, for this reason dataset from the keras library gave by the tensorflow is utilized and certain models are built, to think about the accuracy certain graphs are made and a few parameters are changed to perceive how they influence the model.

INTRODUCTION: -

Before jumping profound into the image classification part, it is important to comprehend the essential ideas of artificial neural networks, deep learning and have a fundamental information on tensorflow. In the accompanying segments I will be giving a brief on these ideas and afterward will dive further into the classification of images. A few strategies have just been acquainted in the market with distinguish and classify images and the one considered best is a convolution neural system model yet this paper will be increasingly engaged towards the fundamental machine learning model and attempt to alter the data in a manner so they can chip away at these models and give us some forecast and we will see which one ends up being the best and why, when classifying images, the hardest part is to extract the relevant features and this is the reason deep learning has the advantage on the grounds that these models can do it successfully. In any case, the inconvenience is that even a little mistake can influence these models in an unfavourable way so the data given to the model ought to be pre-processed appropriately or the forecast won't end up being sufficient.

Artificial Neural Networks-

Neural network is viewed as a recreation of human cerebrum the manner in which our neurons work by responding to the progressions and imparting signs through neurotransmitters similarly, we have displayed modelled nodes and weights to get changed to specific changes.

Fundamental neural network model comprises of three layers:

1. Input Layer: A gathering of input nodes with certain weights and biases we give the inputs to this layer the quantity of nodes taken relies upon the issue portrayal.

2. Hidden Layer: The most significant piece of a neural network where all the tweaking and back propagation happens the quantity of nodes in the hidden layer is reliant on the issue portrayal likewise yet it must be remembered that progressively number of nodes in the hidden layer won't furnish us with increasingly accurate outcomes there is constantly a threshold that shouldn't be crossed.

3.Output Layer: In the event of classification reason there will be generally a solitary node can be more which will anticipate the ideal outcome.

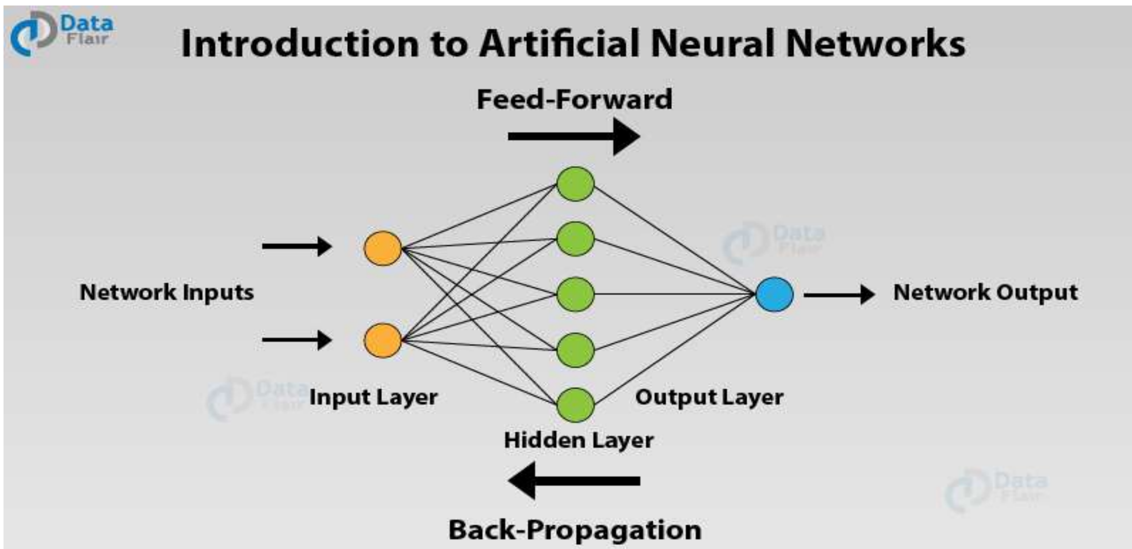


Fig. 1- Artificial Neural Network

The common question that comes to everyone's mind is that how are these simple nodes able to learn things this is where the weights and biases are introduced the data that we provide to these networks are labelled data when it is under processing these nodes try to adjust these weights so that it can correct itself if it predicted the wrong output and after a lot of tweaking of weights and biases it comes up with a model that can work as desired. When we introduce more and more hidden layers, we come to the deep learning part. If we introduce a large number of hidden layers it becomes really difficult to understand how the model actually works and to correct it would be a hectic task. Thus, now we come to the hidden layers part and deep learning in the machine learning field which talks about the exact same stuff and how to build or organize these layers so that it works as desired.

Deep Learning-

At the point when the neural networks have numerous layers which are significant for image classification on account of the huge data that it needs to process and concentrate important features from it we concoct deep learning which is a subfield of machine learning these models can once in a while even outflank human level knowledge and that is the thing that makes it so stunning. These models require a tremendous measure of information and high computational force and thus we can utilize distributed computing techniques to accomplish this. We for the most part actualize deep learning techniques utilizing neural network therefore we additionally call them deep neural networks. One of the notable deep neural networks is a convolutional neural network, which has numerous applications. What makes deep learning taking in not the same as machine learning is that in conventional machine learning techniques on the features were removed physically yet in deep learning the network naturally extricates the features. Deep learning has applications in many fields which incorporate clinical research, mechanized vehicles, picture acknowledgment, mechanical robotization, etc.

TensorFlow-

An open source python library for Machine Learning aficionado, when making neural networks as opposed to making everything from the scratch whether it be a few nodes or weights we are furnished with inbuilt libraries that can carry out the responsibility for us and make our activity simpler. After Google presented the TensorFlow framework in the market, numerous individuals had the option to utilize it without limit and had the option to think of generally excellent ventures. Here likewise we will utilize the TensorFlow libraries to prepare and assemble our model which can do the image classification. The dataset utilized is from keras library called satellite terrain dataset which has around sixty thousand sources of info and ten thousand for validations we are going to train our model on the equivalent.

Keras-

Keras is a moderate Python library for deep learning that can run on Theano or TensorFlow. It was created to make actualizing deep learning models as quick and simple as feasible for research and development. It can consistently execute on GPUs and CPUs given the fundamental systems. It is discharged under the lenient MIT permit. Keras was created and kept up by François Chollet, a Google engineer utilizing four core values:

- Measured quality: A model can be comprehended as a succession or a chart alone. All the worries of a profound learning model are discrete segments that can be consolidated in discretionary manners.
- Moderation: The library gives sufficiently only to accomplish a result, no nonsense and amplifying meaningfulness.
- Extensibility: New parts are purposefully simple to include and use inside the framework, expected for specialists to preliminary, and investigate new thoughts.
- Python: No different model documents with custom record designs. Everything is local Python.

RELATED WORK: -

Parcel of preferences can be made in remote detection of photos without the person's mediation. The remotely sensed picture is computerized depictions of the Planet, and facilitates the practice of those within pieces, through using that, positions that cannot be seen through the remote detection pictures. Each pixel speaks in a region of the earth in a remotely sensed image information in a given area.

If it is not possible for a pixel to meet a certain set of criteria, the pixel is then classified into a class which compared with those criteria. This is alluded to as a system with pictures. Based upon photos crude for example, pixel-based and object-based technologies can be obtained in two main classifications. Techniques centred on pixels describe individual pixels, without any region or field spatial details being taken into account. Additionally, approaches focused on Article / Region are equipped for the significance with high goals that irritate the grouping phase for most pixel dependent strategies.

Contingent on the type of data extracted from the first details, groups will be linked to the identified highlights on the field. One example is a land distribution map, displaying forest, open property, plain, urban areas, etc. An organized image is. A pixel can refer to a mix of class couches, class shifts, or other complicated instances which cannot properly be represented in a single class while detecting symbolizes from a remote. Determining the degree of the vegetation lists is important if we are to talk about the land used and the rural level in the relevant district.

For this reason, a remote detection image is taken, LANDSAT picture is taken in this research and it is ready to classify the used property. In the planning of the first picture of LANDSAT, concussion is tested. The requisite highlights are segregated from this photo. Diverse highlights such as landscape data, utilized property, woodland and unused land are listed for this extraction. Following the quotation from the illustration, order calculations are rendered to achieve distinguishing arrangements, formulas of fuzzy are added for the described picture in this KNN, SVM. Ordinary controlled image order depends on the processing of data, which is transiently in line with the images used (destinations with immediate expectations of land expansion).

In multivariate accurate analyses, the processing of knowledge and multi-ghastly satellite data for specific destinations is used to generate a precious pattern, defined as' irrevolutionary signage,' used for grouping the satellite image to the classroom of land distribution. The preparing of the material, however, is generally not accessible for most photos in a time and cannot be effectively acquired, in general, for more developed photos.

The transient signature increase has produced favoured results than the expansion of spatial signatures especially where variability is diminished over the past years with radiometric standards (or modification, however, there has been little awareness of the ordinary legality of the way the expansion of signatures has been treated.

WHY THIS TOPIC?

At the time we have option to select a topic for term paper. We had few questions like how this topic help us for future studies, what is the research scope of this field in upcoming few year, how much research has been already conducted or paper publish in the journals.

After looking all the above parameters, we decided that it has something fascinating and we can put some extra effort to explore it deeper based on previous researches and the studies. At the same time this topic will also give few challenges to our algorithm making and coding skill while implementation of the written algorithm.

CHALLENGES: -

- Dataset : Selection of dataset most import when we are suppose to conduct some study on any algorithm, if the chosen dataset is not random data then we can face the issues with the accuracy measures because it may be consider that the output provided by the algorithm is biased and that can be convert into failure as a result.

- Algorithm selection: Suppose we have selected some dataset, it has some field value as null, then which method we have to use like mean, median, mode or average to feed the value at null space. Which column is of our requirement and which have to discard from our dataset so that it can not affect the overall result?

- Accuracy: It is the most Important factor in the field of machine learning, the whole effort result shown as accuracy of our prediction, it depends upon many factors like input data, target data, algorithm used in it, data cleaning, making data randomize.

METHODOLOGY: -

Terrain classification architectural model is shown below in picture. In this picture as we can see there 4 stages of project proceeding:

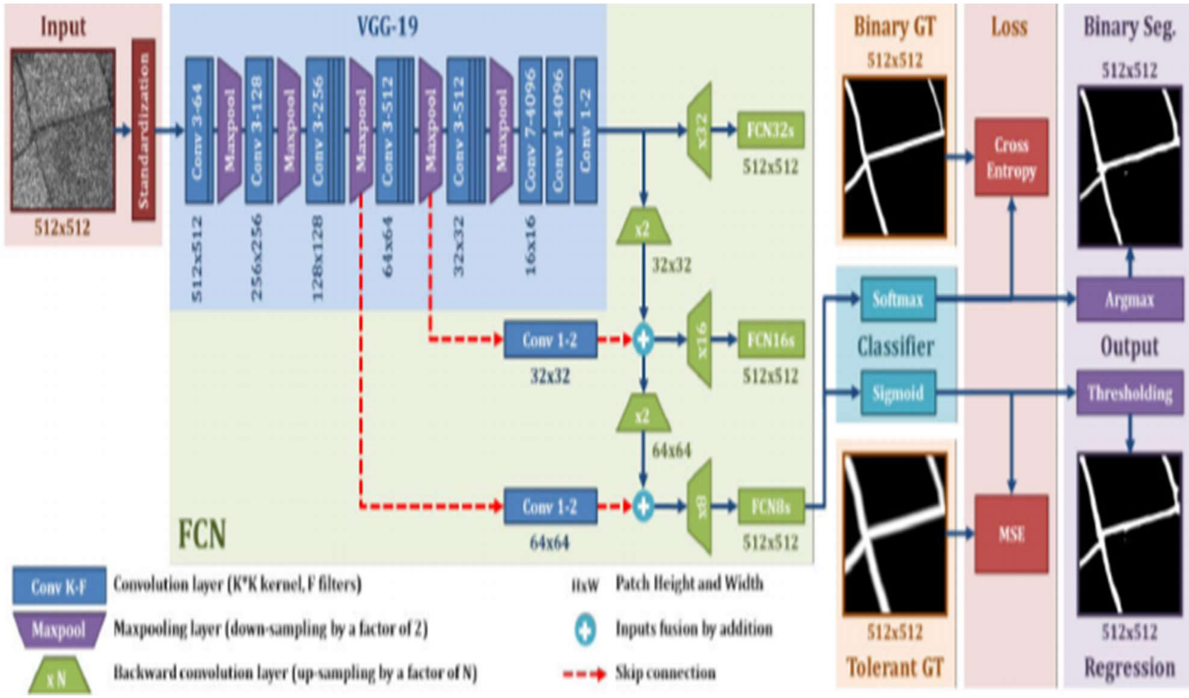


Fig. 2: Road Segmentation in SAR Satellite Images with Deep Fully-Convolutional Neural Networks

Stage 1: Collection of dataset and preparing it.

Stage 2: Developing a CNN based Terrain classification model.

Stage 3: Segmentation of the images and training on the built dataset.

Stage 4: Implementation and Analysis

If the implementation and the analysis reach up to the mark then we can use it real time scenario for further analysis and development.

IMPLEMENTATION AND TRAINING: -

The model has been actualized and tried utilizing the Keras library with TensorFlow as backend. Keras is a profound learning library that has picked up prominence in the previous barely any years and can utilize either TensorFlow or Theano as backend. The calculation was structured and executed in the accompanying manner: Initial, a scaled down group of pictures (rearranged arbitrarily) is produced from the preparation set. The pictures are of measurements 28x28. The picture increase method at that point utilized on those created pictures. At last, they are taken care of to the learning calculation. The selection of measurements 28x28 for the information pictures was chosen because of the way that a street in the dataset ordinarily contains roughly 40 pixels of width or tallness. We included 16 pixels each side of the landscape so as to consider the setting when preparing, making up a sum of 72.

Moreover, the misfortune work utilized is the twofold cross entropy work, which is an uncommon instance of the unmitigated cross entropy and is appropriate for grouping undertakings as indicated by inquire about. At long last, the model uses Early Stopping call back which stops the realizing when the model quits improving after 10 stages and ReduceLROnPlateau that lessen the learning rate when the model quits improving after 4 stages. This guarantees the model attempts to decrease the learning rate at any rate twice before for all time halting.

When all is said in done, models show to frequently profit by diminishing the learning rate by a factor of 2-10 to learn stagnates, which have additionally been powerful for our subsequent model. The quantity of age was exactly picked to be 150, with 1500 stages for every age. The preparation running the preparation on a nearby PC with less processing power.

DIFFERENT MODEL EVALUATION ON SAT4 SATELLITE TERRAIN DATASET:

Using different activation functions, I have created three models and set the epoch equals five and got the following results. My basic model has 1000 input nodes a 28x28 matrix image can be given as input to these nodes a hidden layer of 128 nodes and have 9 output nodes on which I have applied the SoftMax function. The model is trained on ninety-eight thousand nine hundred and ninety-nine input, images and predicted on one thousand validation images. While working with each model there are various observations and analysis.

SoftMax: -

```
model = Sequential([
    Dense(4, input_shape=(3136,), activation='softmax'),
#     Dense(4, input_shape=(3136,), activation='sigmoid'),
#     Dense(4, input_shape=(3136,), activation='relu')
])
```

```
X_train = X_train/255
```

```
# vanilla artificial neural network
# multi class classifcation => categorical_crossentropy
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
model.fit(X_train,Y_train,batch_size=32, epochs=5, verbose=1, validation_split=0.01) # more epoch may give the more accuracy
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 4)                 12548
=================================================================
Total params: 12,548
Trainable params: 12,548
Non-trainable params: 0
_________________________________________________________________
Train on 98999 samples, validate on 1000 samples
Epoch 1/5
98999/98999 [==============================] - 20s 207us/step - loss: 0.7581 - accuracy: 0.6880 - val_loss: 0.7650 - val_accura
cy: 0.6150
Epoch 2/5
98999/98999 [==============================] - 19s 193us/step - loss: 0.6707 - accuracy: 0.7203 - val_loss: 0.7091 - val_accura
cy: 0.6780
Epoch 3/5
98999/98999 [==============================] - 19s 191us/step - loss: 0.6640 - accuracy: 0.7208 - val_loss: 0.5954 - val_accura
cy: 0.7660
Epoch 4/5
98999/98999 [==============================] - 19s 193us/step - loss: 0.6528 - accuracy: 0.7271 - val_loss: 0.7567 - val_accura
cy: 0.6770
Epoch 5/5
98999/98999 [==============================] - 19s 195us/step - loss: 0.6419 - accuracy: 0.7293 - val_loss: 0.6137 - val_accura
cy: 0.7450
```

The following shows the training phase of SoftMax model the accuracy starts from 61% and increases up to 74% also when the model is evaluated on the test data it shows 74% of accuracy there could be a little bit of overfitting but it can be ignored.

Sigmoid: -

The following shows the training phase of Sigmoid model the accuracy starts from 38% and increases up to 38% also when the model is evaluated on the test data it shows 38% of accuracy there could be a little bit of overfitting but it can be ignored.

```
model = Sequential([
#     Dense(4, input_shape=(3136,), activation='softmax'),
    Dense(4, input_shape=(3136,), activation='sigmoid'),
#     Dense(4, input_shape=(3136,), activation='relu')
])
```

```
X_train = X_train/255
```

```
# vanilla artificial neural network
# multi classs classififcation => categorical_crossentropy
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
model.fit(X_train,Y_train,batch_size=32, epochs=5, verbose=1, validation_split=0.01) # more epoch may give the more accuracy
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_9 (Dense)              (None, 4)                 12548
=================================================================
Total params: 12,548
Trainable params: 12,548
Non-trainable params: 0
_________________________________________________________________
Train on 98999 samples, validate on 1000 samples
Epoch 1/5
98999/98999 [==============================] - 37s 375us/step - loss: 1.3540 - accuracy: 0.3559 - val_loss: 1.3388 - val_accura
cy: 0.3870
Epoch 2/5
98999/98999 [==============================] - 19s 196us/step - loss: 1.3469 - accuracy: 0.3560 - val_loss: 1.3357 - val_accura
cy: 0.3870
Epoch 3/5
98999/98999 [==============================] - 21s 207us/step - loss: 1.3445 - accuracy: 0.3560 - val_loss: 1.3336 - val_accura
cy: 0.3870
Epoch 4/5
98999/98999 [==============================] - 20s 204us/step - loss: 1.3419 - accuracy: 0.3560 - val_loss: 1.3315 - val_accura
cy: 0.3870
Epoch 5/5
98999/98999 [==============================] - 21s 210us/step - loss: 1.3391 - accuracy: 0.3560 - val_loss: 1.3279 - val_accura
cy: 0.3870
```

ReLu: -

The following is the training phase it is well observed that the accuracy rate gradually increases on the training data which is good but after we evaluate the model on the test data we can also see that there is a three percent difference of accuracy rate from which we can also imply that maybe the model is overfitting instead of generalizing the things it is trying to memorize the features.

```
model = Sequential([
#     Dense(4, input_shape=(3136,), activation='softmax'),
#     Dense(4, input_shape=(3136,), activation='sigmoid'),
    Dense(4, input_shape=(3136,), activation='relu')
])
```

```
X_train = X_train/255
```

```
# vanilla artificial neural network
# multi classs classififcation => categorical_crossentropy
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
model.fit(X_train,Y_train,batch_size=32, epochs=5, verbose=1, validation_split=0.01) # more epoch may give the more accuracy
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_11 (Dense)             (None, 4)                 12548
=================================================================
Total params: 12,548
Trainable params: 12,548
Non-trainable params: 0
_________________________________________________________________
Train on 98999 samples, validate on 1000 samples
Epoch 1/5
98999/98999 [==============================] - 18s 179us/step - loss: 7.4869 - accuracy: 0.3370 - val_loss: 6.7772 - val_accura
cy: 0.3870
Epoch 2/5
98999/98999 [==============================] - 20s 201us/step - loss: nan - accuracy: 0.2984 - val_loss: nan - val_accuracy: 0.
2280
Epoch 3/5
98999/98999 [==============================] - 19s 192us/step - loss: nan - accuracy: 0.2622 - val_loss: nan - val_accuracy: 0.
2280
Epoch 4/5
98999/98999 [==============================] - 20s 204us/step - loss: nan - accuracy: 0.2622 - val_loss: nan - val_accuracy: 0.
2280
Epoch 5/5
98999/98999 [==============================] - 20s 200us/step - loss: nan - accuracy: 0.2622 - val_loss: nan - val_accuracy: 0.
2280
```

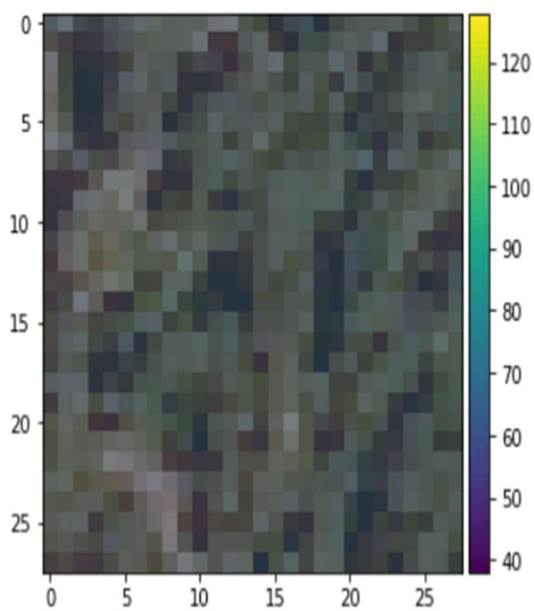
Now to know the generalization factor we will train the model on each epoch from 1-5 and observe what happens to train accuracy and test accuracy.

Training Data: -

We should provide the particular data set into the code to get the prediction for the provided data. The following is the training phase it is well observed that the accuracy rate gradually increases on the training data which is good but after we evaluate the model on the test data we can also see that there is a three percent difference of accuracy rate from which we can also imply that maybe the model is overfitting instead of generalizing the things it is trying to memorize the features.

```
 ix = 10  #training data and the input
imshow(np.squeeze(X_train_img[ix,:,:,0:3]).astype('int32')) #Only seeing the RGB channels
plt.show()
#Tells what the image is
if Y_train[ix,0] == 1:
    print ('Barren Land')
elif Y_train[ix,1] == 1:
    print ('Trees')
elif Y_train[ix,2] == 1:
    print ('Grassland')
else:
    print ('Other')
```

```
C:\Users\shakt\AppData\Local\Continuum\anaconda3\lib\site-packages\skimage\io\_plugins\matplotl
w image data range; displaying image with stretched contrast.
  warn("Low image data range; displaying image with "
```



The above image shows us the whatever we give as an input it will provide the following output.

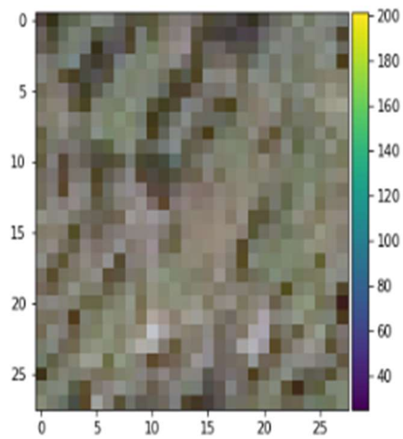
RESULT:

With the above discoveries, various models have been tried on our dataset, with pattern having the most minimal outcomes and utilized for examination purposes. At that point, a few models of CNN have likewise been actualized to perceive how they contrast. The thing that matters is including and expelling at least one of the improvement systems talked about in the execution shows the outcomes for every engineering with the prescient outcomes with a probabilistic methodology.

```python
ix = 10 #input number in range 0 99999 // 99999 last set of data in dataset
imshow(np.squeeze(X_train_img[99999-(1000-ix),:,:,0:3]).astype('int32')) #Only seeing the RGB channels
plt.show()
#give the image is?
print ('Prediction:\n{:.2f}% Probability barren land,\n{:.2f}% Probability trees,\n{:.2f}% Probability grassland,\n{:.2f}% P

print ('Final Conclusion: ',end='')
if Y_train[99999-(1000-ix),0] == 1:
    print ('Barren Land')
elif Y_train[99999-(1000-ix),1] == 1:
    print ('Trees')
elif Y_train[99999-(1000-ix),2] == 1:
    print ('Grassland')
else:
    print ('Other')
```

```
C:\Users\shakt\AppData\Local\Continuum\anaconda3\lib\site-packages\skimage\io\_plugins\matplotlib_plugin.py:75: UserWarning: Lo
w image data range; displaying image with stretched contrast.
  warn("Low image data range; displaying image with "
```




```
Prediction:
0.34% Probability barren land,
39.16% Probability trees,
51.03% Probability grassland,
9.48% Probability other
```

<u>Why CNN is better?</u>

Well there is constantly a cut-off to which we can foresee with a typical model with straightforward nodes and weights, it likewise has certain disadvantages on the off chance that we give a similar image to the model yet in a zoomed, flipped, or reflected in vertical or flat direction it won't have the option to anticipate it accurately thus we need a convolutional neural network to understand the issue utilizing feature mapping and pooling methods we can sum up the model considerably further and concoct a model that can predict or classify all the more accurately.

It begins with straightforward features and progressively focuses around an ever-increasing number of subtleties yet to make it work you need to expand your dataset in the training stage itself that is on the off chance that you have an image as an input zoom it or flip it or mirror it and make more information sources and give them during the training stage.

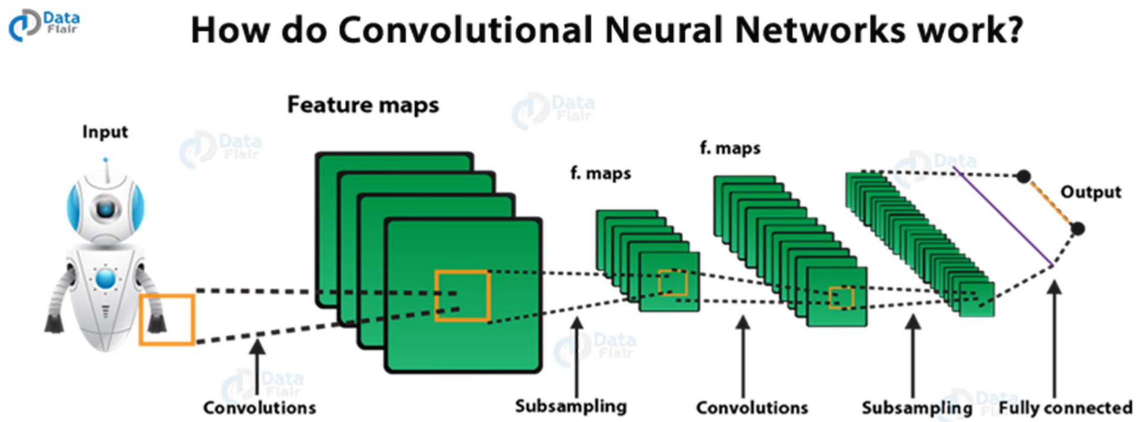


Fig.3 Wornking of CNN

CONCLUSION:

In this paper I have given a brief inside look into image classification using the models that I have created have also compared certain observations that I have obtained it is important to visualize the results to come up with certain conclusions and know the trend of how in each iteration the accuracy rates changes and if it is feasible to consider those results or is it making the model worse by simply overfitting upon the data. Also, whether tweaking certain parameters really have an impact on the model itself. The three-activation functions that I have considered are softmax, sigmoid, and relu there are several other activation functions but I find these three to be the most common and easier to understand from the observations I have come to the conclusion that for this particular dataset softmax activation function with multiple hidden layer suits best and can give you an accuracy increase the accuracy even more I personally believe that we have to implement the convolutional neural network along with augmenting the dataset.

REFRENCES:

1. Muhammad, S., Aziz, G., Aneela, N. and Muhammad, S. 2012. “Classification by Object Recognition in SatelliteImages by using Data Mining”. In Proc. Proceedings of the World Congress on Engineering (WCE 2012), Vol I, July 4 - 6, London, U.K.
2. Chaichoke, V., Supawee, P., Tanasak, V. and Andrew, K, S. 2011. "A Normalized Difference Vegetation Index (NDVI) Time-Series of Idle Agriculture Lands: A Preliminary Study", Engineering Journal. Vol. 15, Issue 1, pp. 9-16.
3. Zheng, X., Sun, X., Fu, K. and Hongqi Wang, 2013. “Automatic Annotation of Satellite Images via Multifeature Joint Sparse Coding With Spatial Relation Constraint”, IEEE Geoscience and Remote Sensing Letters, VOL. 10, NO. 4, JULY 2013, pp.652-656.
4. Anders Karlsson, 2003. “Classification of high resolution satellite images”, August 2003, available at http://infoscience.epfl.ch/record/63248/files/TPD_Karlss on.pdf..
5. Pattern Recognition Scheme for Large-Scale Cloud Detection Over Landmarks Adrián Pérez-Suay ; Julia Amorós-López ; Luis Gómez-Chova ; Jordi Muñoz-Marí ; Dieter Just ; Gustau Camps-Valls IEEE 2018.
6. Soliman, O, S. and Mahmoud, A.S., 2012. “A classification system for remote sensing satellite images using support vector machine with non-linear kernel functions”, In proc. 8th International Conference on Informatics and Systems (INFOS), IEEE, 14-16 May 2012, pp.BIO-181,BIO-187, Cairo.
7. Horning, N. 2004. “Land cover classification methods”, Version 1.0. American Museum of Natural History, Center for Biodiversity and Conservation. Available at http://biodiversityinformatics.amnh.org.

8. Murugeswari, P. and Manimegalai, D. 2012. "Color Textured Image Segmentation Using ICICM – Interval Type-2 Fuzzy C-means Clustering Hybrid Approach", Engineering Journal, Vol. 16, No. 5, pp. 115-126.
9. Al-Ahmadi, F, S. and Hames, A, S. 2009. "Comparison of Four Classification Methods to Extract Land Use and Land Cover from Raw Satellite Images for Some Remote Arid Areas, Kingdom of Saudi Arabia", Journal of King Abdulaziz University-Earth Sciences, Vol. 20, No.1, pp: 167-191.
10. Ahmed, R., Mourad, Z., Ahmed, B, H. and Mohamed, B. 2009. "An Optimal Unsupervised Satellite Image Segmentation Approach Based on Pearson System and kMeans Clustering Algorithm Initialization", International Science Index, Vol. 3, No. 11, pp. 948-955.