CHANGE DETECTION AND IMAGE CLASSIFICATION IN IMAGES

Guided by:
Dr. K. R. Manjula, Associate Professor, School of Computing

123003058 - Deepak K
123015092 - Shiv Mohan L
123015121 - Yaredla Sai Charith Reddy

Base paper details

- Tittle: Unsupervised Change Detection in Landsat Images with Atmospheric Artifacts
- Authors: Kusetogullari Huseyin and Yavariabdi Amir
- Journal: Hindawi Journals/Mathematical Problems in Engineering
- Year of publication: 2018
- Indexing: Scopus
- Base-paper URL:Unsupervised Change Detection in Landsat Images with Atmospheric Artifacts: A Fuzzy Multiobjective Approach (hindawi.com)

CONTENT

- Base Paper Details
- Abstract
- Introduction
- Problem Statement
- Literature Review
- Methodology
  - Data Collection
  - Unsupervised Learning
  - Supervised Learning
- Result analysis
- Conclusion and future works
- Workflow
- References

ABSTRACT

- In remote sensing context, many automatic change detection algorithms have been proposed to obtain changed and unchanged pixels between images.
- Change detection helps in identifying the differences in satellite images taken at different points in time.
- The proposed method uses a photoelectric invariant technique,along with hybrid wavelet transform and mean shift clustering.
- Then a fusion method is used to get final change mask which is further used for the finding finding the change in real time.
- Also, The input satellite images are classified into using a customized labeled dataset.

INTRODUCTION

- The proposed method uses a photoelectric invariant technique,along with hybrid wavelet transform and mean shift clustering combined with an optimization technique.
- Various Clustering techniques are implemented to find out the most optimized change detection result
- Image Classification is typically achieved using machine learning algorithms, such as deep neural networks.
- Here, to classify satellite images Residual Network model(ResNet-50) is used.
- Its effective and efficient architecture delivers high accuracy with very few parameters.

PROBLEM STATEMENT

- Comparative analysis to find the most optimized method in finding the change in multi temporal satellite images.
- Classification of satellite images into a desired labeled dataset.

LITERATURE REVIEW

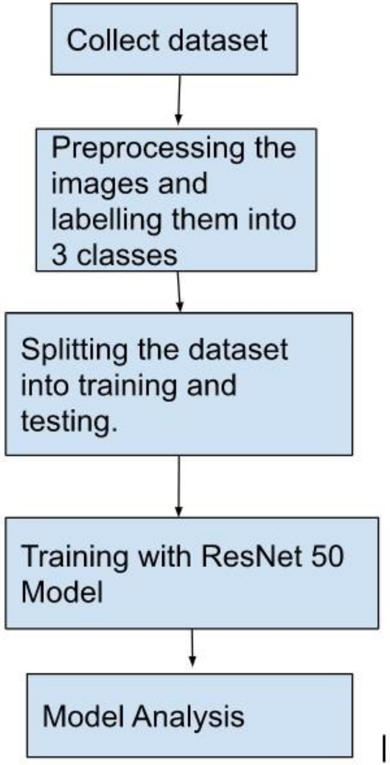
S.No.	Title	Journal	Work
1	Agriculture change detection model using remote sensing images and GIS	IEEE Xplore	Changes in agriculture detected using Landsat images
2	Fusion of difference images for change detection over urban areas	IEEE Xplore	Detecting spectral changes through fusion techniques merging multiple difference images in order to understand urbanization.
3	A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications	Hindawi	Analyzing the usage and applications of Particle Swarm Optimization.
4	Land Use and Land Cover using Deep Learning	IEEEXplore	Using Deep learning algorithm in classifying images.
5	Segmentation and histogram generation using the HSV color space for image retrieval	IEEE Xplore	Analysis of HSV colorspace with emphasis on visual perception of variation in hue,saturation and intensity values.

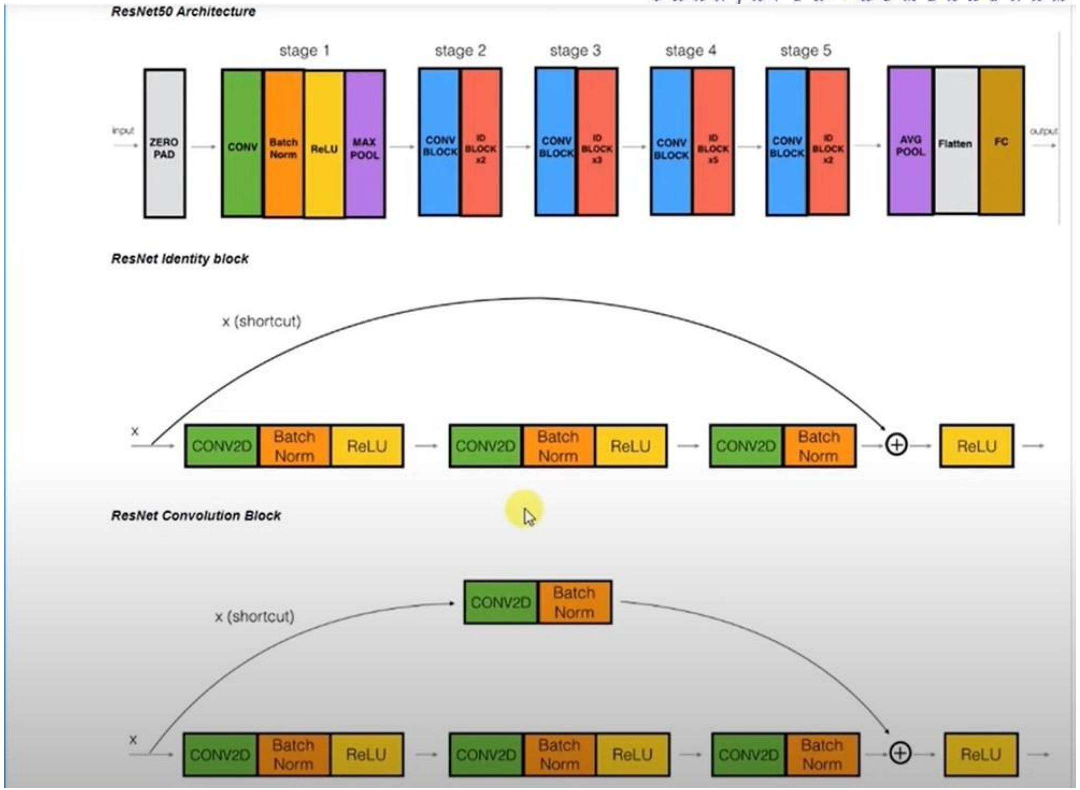
PROPOSED METHOD

The proposed method uses the following steps:

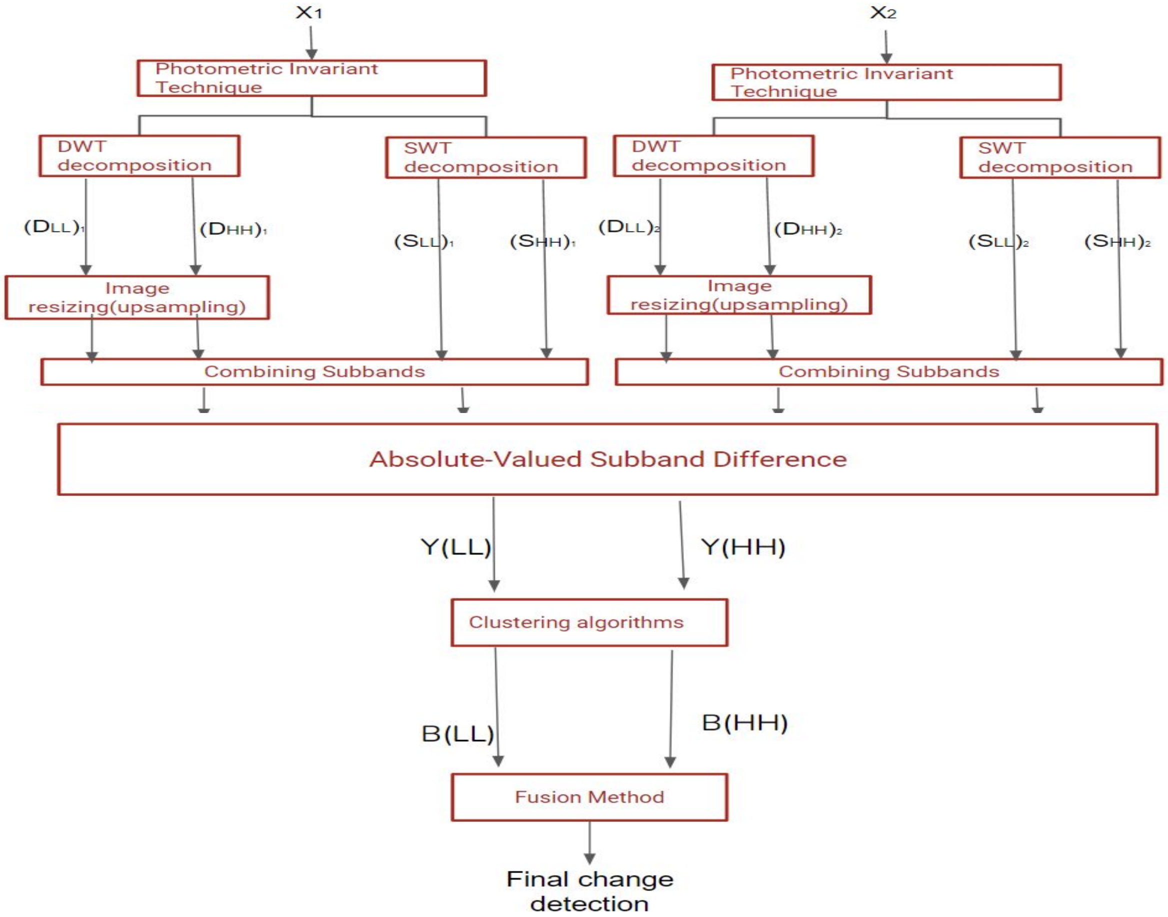
- Firstly, a photometric invariant technique is used to transform the Landsat images from RGB to HSV colour space.
- A hybrid wavelet transform based on Stationary and Discrete Wavelet Transforms is applied.
- After that, mean shift clustering method is applied to the subband difference images, computed using the absolute-valued difference technique.
- Then, the proposed method optimizes using PSO to evaluate changed and unchanged regions of the smoothed difference images separately.
- Finally, a fusion approach based on connected component with union technique is proposed to fuse two binary masks to estimate the final solution.
- The input images are also classified into a labeled dataset using ResNet model and its accuracy is predicted.

METHODOLOGY





Workflow



X1
X2
Photometric Invariant Technique
Photometric Invariant Technique
DWT decomposition
SWT decomposition
DWT decomposition
SWT decomposition
(DLL)1
(DHH)1
(SLL)1
(SHH)1
(DLL)2
(DHH)2
(SLL)2
(SHH)2
Image resizing(upsampling)
Image resizing(upsampling)
Combining Subbands
Combining Subbands
Absolute-Valued Subband Difference
Y(LL)
Y(HH)
Clustering algorithms
B(LL)
B(HH)
Fusion Method
Final change detection

(DATASET-Clustering)

*INPUT IMAGES: SASTRA University (10^o*43’45.66’’N 79^{o}01’08.01”E) taken from Google Earth Pro

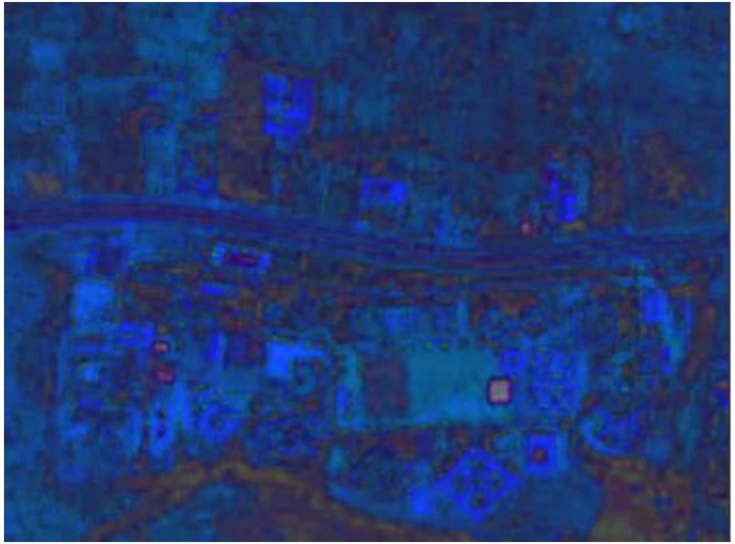
2013

2023

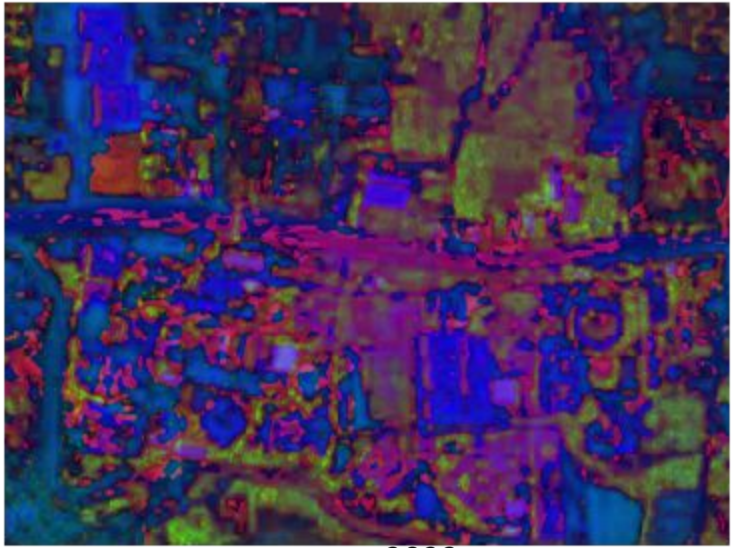
DATASET-Classification

- The dataset consists of pre processed images collected over a 10 year period.
- In pre processing, the input images are cleared of clouds and each image is divided into 192 images of 64x64 size.
- Each image is categorized into one of the 3 labels(i.e Residential, Industry and Herbaceous Vegetation).
- DATASET

ColorSpace Transformation (Unsupervised Learning)



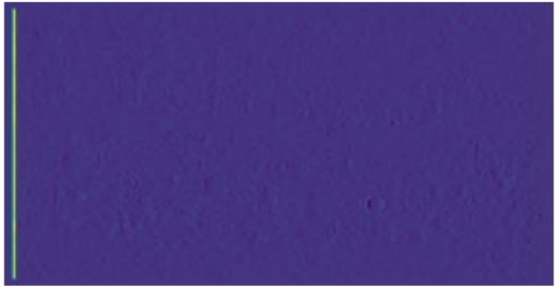
2013



2023

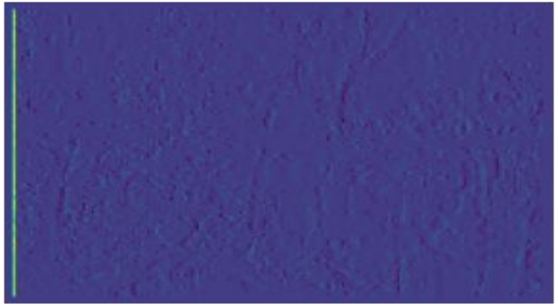
Discrete Wavelet Transform

2013-LL



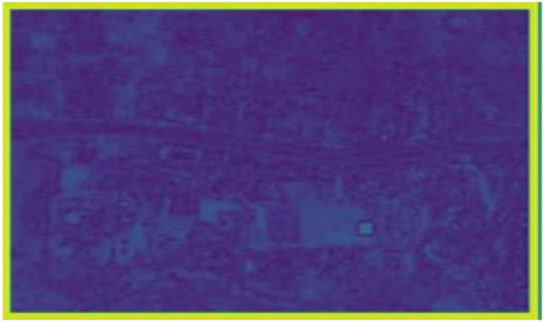
2013-HH

2023-LL

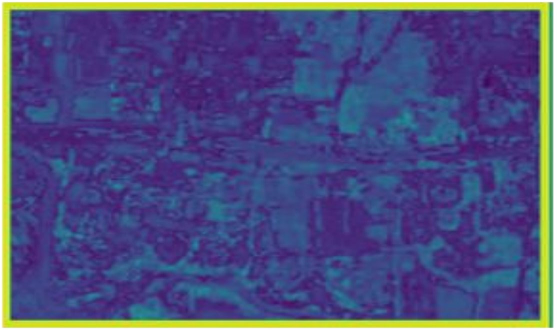


2023-HH

Stationary Wavelet Transform

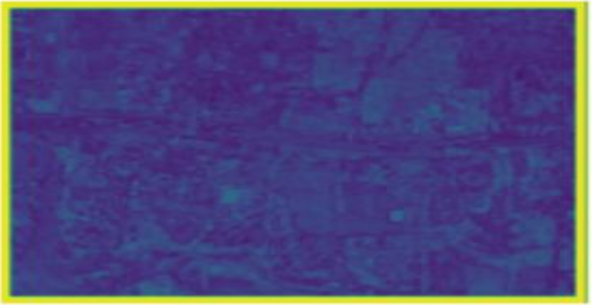

2013-LL


2013-HH


2023-LL

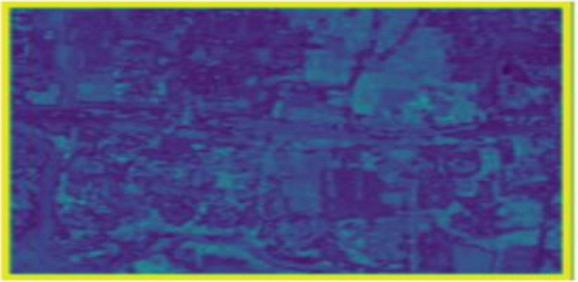

2023-HH

Combining Subbands



2013-LL

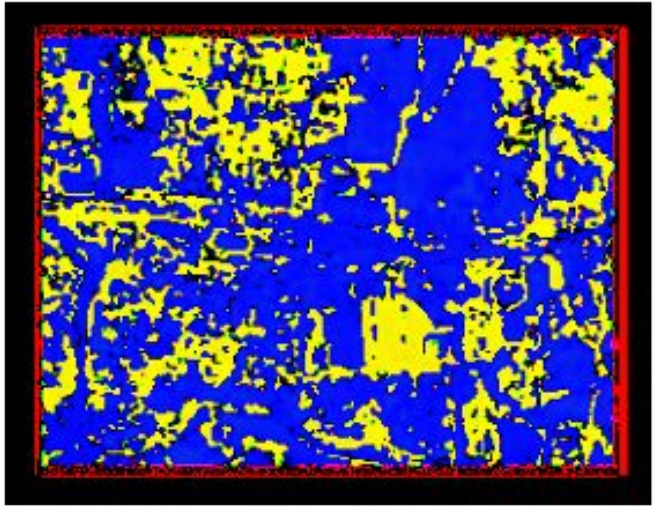
2013-HH



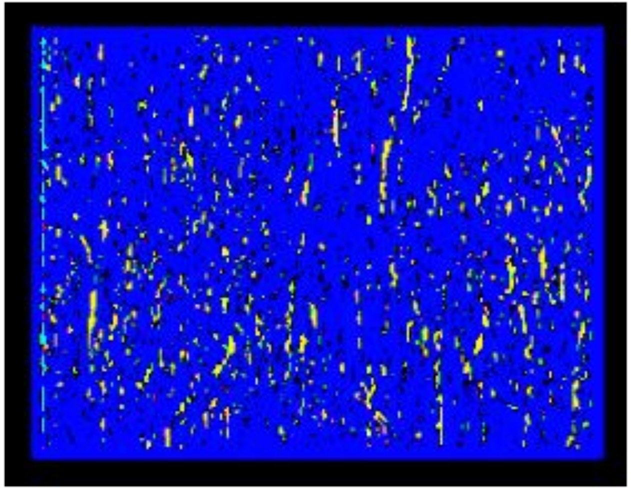
2023-LL

2023-HH

Absolute Valued Subband



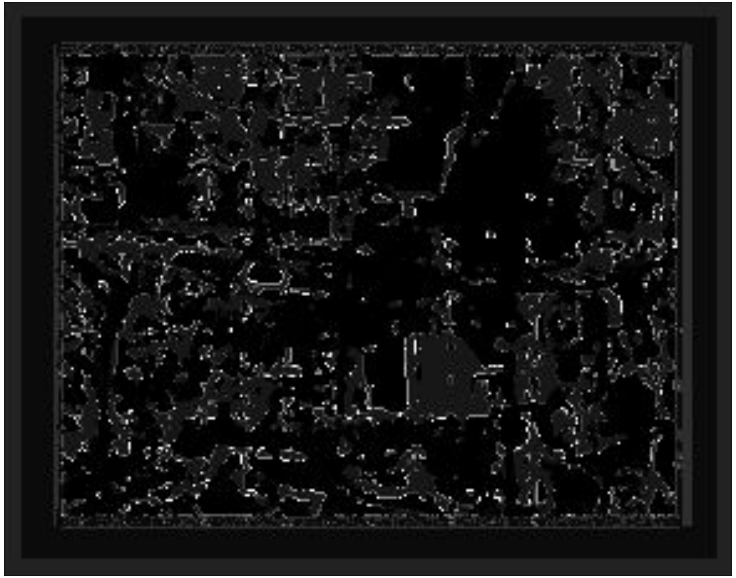
LL



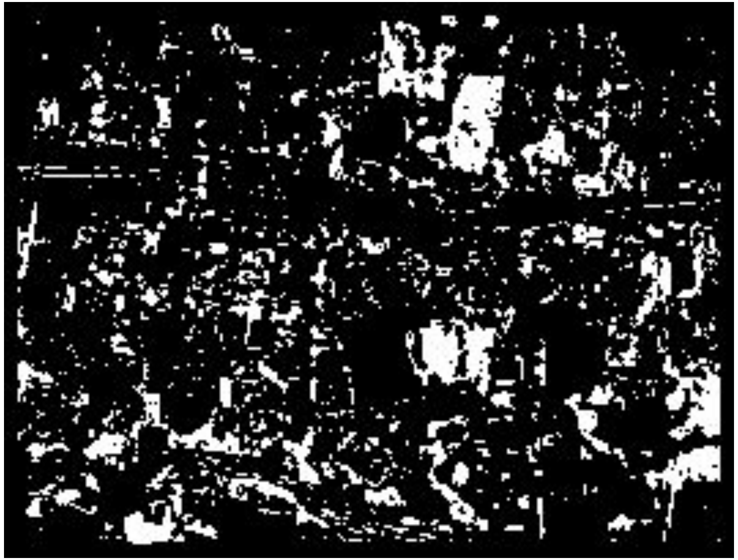
HH

K-Means

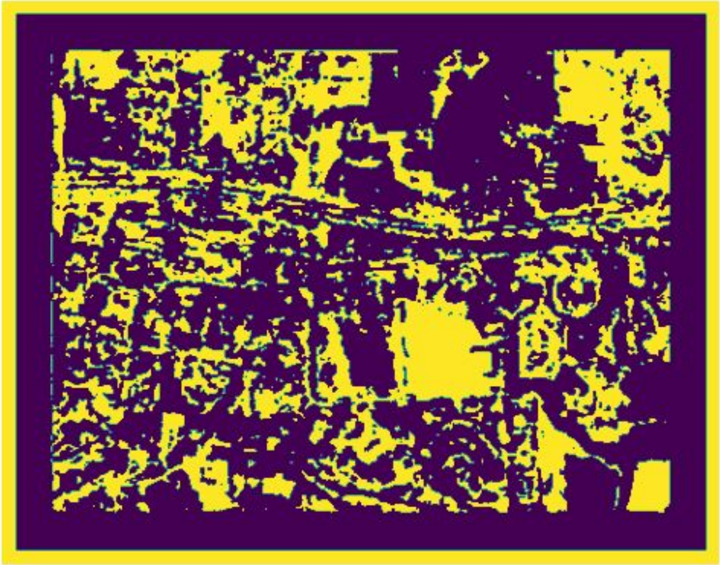
Mean Shift Clustering



DT-CWT



Particle Swarm Optimization

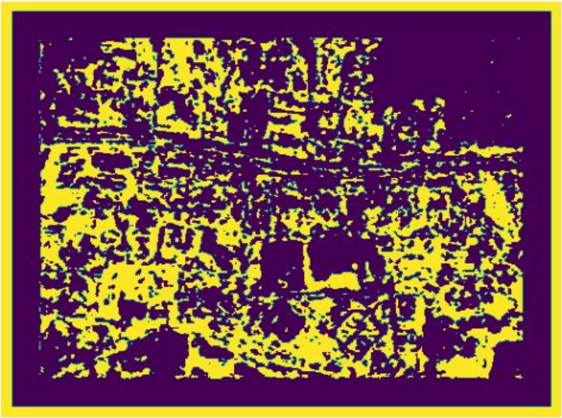


2013-2018

2013

2018

INPUT



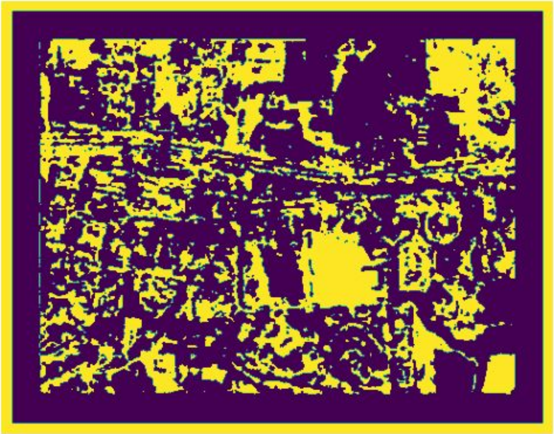
OUTPUT(change)

2018-2023

2018

2023

INPUT

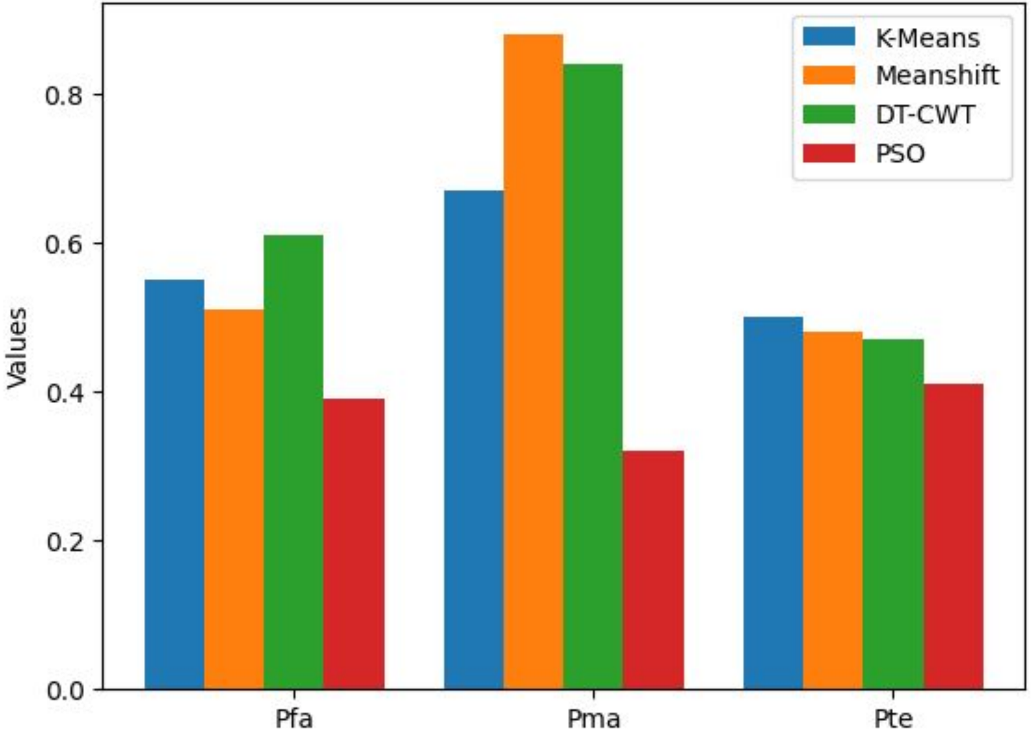


OUTPUT(change)

Performance analysis

	K-Means	Mean-shift	DT-CWT	**PSO**
Pfa(false alarm)	0.65	0.5	0.56	**0.41**
Pmd(missed detection)	0.75	0.87	0.85	**0.38**
Pte(total error)	0.61	0.61	0.58	**0.53**

Performance analysis



Model training-(Classification) (Supervised Learning) ResNet50 Model

```
Epoch 1/10
101/101 [==============================] - 2469s 24s/step - loss: 1.6674 - accuracy: 0.6438 - val_loss: 0.9265 - val_accuracy: 0.6335
Epoch 2/10
101/101 [==============================] - 631s 6s/step - loss: 0.9933 - accuracy: 0.6692 - val_loss: 1.0030 - val_accuracy: 0.6907
Epoch 3/10
101/101 [==============================] - 626s 6s/step - loss: 0.7909 - accuracy: 0.6962 - val_loss: 0.8407 - val_accuracy: 0.6683
Epoch 4/10
101/101 [==============================] - 621s 6s/step - loss: 0.7274 - accuracy: 0.7068 - val_loss: 1.0872 - val_accuracy: 0.6994
Epoch 5/10
101/101 [==============================] - 605s 6s/step - loss: 0.6406 - accuracy: 0.7316 - val_loss: 0.6237 - val_accuracy: 0.7826
Epoch 6/10
101/101 [==============================] - 611s 6s/step - loss: 0.7361 - accuracy: 0.7148 - val_loss: 0.7497 - val_accuracy: 0.6211
Epoch 7/10
101/101 [==============================] - 616s 6s/step - loss: 0.8750 - accuracy: 0.6922 - val_loss: 0.7452 - val_accuracy: 0.7031
Epoch 8/10
101/101 [==============================] - 595s 6s/step - loss: 0.7010 - accuracy: 0.7154 - val_loss: 0.9345 - val_accuracy: 0.7665
Epoch 9/10
101/101 [==============================] - 591s 6s/step - loss: 0.6692 - accuracy: 0.7393 - val_loss: 0.8542 - val_accuracy: 0.7043
Epoch 10/10
101/101 [==============================] - 610s 6s/step - loss: 0.6987 - accuracy: 0.7278 - val_loss: 0.4850 - val_accuracy: 0.7727
```

ResNet50 model with optimiser(adam)

```
history = model.fit(train_dataset, validation_data=test_dataset, epochs=10, batch_size=32)
```

```
Epoch 1/10
101/101 [==============================] - 632s 6s/step - loss: 1.2980 - accuracy: 0.6631 - val_loss: 1.5225 - val_accuracy: 0.6484
Epoch 2/10
101/101 [==============================] - 351s 3s/step - loss: 0.8470 - accuracy: 0.6913 - val_loss: 0.9988 - val_accuracy: 0.7006
Epoch 3/10
101/101 [==============================] - 351s 3s/step - loss: 0.6660 - accuracy: 0.7387 - val_loss: 0.6026 - val_accuracy: 0.7739
Epoch 4/10
101/101 [==============================] - 350s 3s/step - loss: 0.6367 - accuracy: 0.7418 - val_loss: 0.8202 - val_accuracy: 0.6994
Epoch 5/10
101/101 [==============================] - 349s 3s/step - loss: 0.6299 - accuracy: 0.7443 - val_loss: 0.5192 - val_accuracy: 0.8050
Epoch 6/10
101/101 [==============================] - 352s 3s/step - loss: 0.5493 - accuracy: 0.7743 - val_loss: 0.4986 - val_accuracy: 0.7553
Epoch 7/10
101/101 [==============================] - 353s 3s/step - loss: 0.5506 - accuracy: 0.7855 - val_loss: 0.7973 - val_accuracy: 0.7776
Epoch 8/10
101/101 [==============================] - 352s 3s/step - loss: 0.5611 - accuracy: 0.7932 - val_loss: 0.6385 - val_accuracy: 0.7565
Epoch 9/10
101/101 [==============================] - 357s 4s/step - loss: 0.4634 - accuracy: 0.8180 - val_loss: 0.5272 - val_accuracy: 0.7665
Epoch 10/10
101/101 [==============================] - 352s 3s/step - loss: 0.4975 - accuracy: 0.8149 - val_loss: 0.5434 - val_accuracy: 0.7901
```

DenseNet model (without optimiser)

```
print("Total time taken by the model:", total_time, "seconds")
```

```
Epoch 1/10
101/101 [==============================] - 2688s 26s/step - loss: 0.7034 - accuracy: 0.7415 - val_loss: 0.4657 - val_accuracy: 0.8211
Epoch 2/10
101/101 [==============================] - 11s 106ms/step - loss: 0.4126 - accuracy: 0.8149 - val_loss: 0.4958 - val_accuracy: 0.8224
Epoch 3/10
101/101 [==============================] - 11s 105ms/step - loss: 0.3202 - accuracy: 0.8571 - val_loss: 0.6496 - val_accuracy: 0.7826
Epoch 4/10
101/101 [==============================] - 11s 106ms/step - loss: 0.2427 - accuracy: 0.9014 - val_loss: 0.9986 - val_accuracy: 0.7739
Epoch 5/10
101/101 [==============================] - 10s 102ms/step - loss: 0.1984 - accuracy: 0.9194 - val_loss: 0.9814 - val_accuracy: 0.7540
Epoch 6/10
101/101 [==============================] - 11s 109ms/step - loss: 0.1550 - accuracy: 0.9402 - val_loss: 1.4511 - val_accuracy: 0.7466
Epoch 7/10
101/101 [==============================] - 11s 106ms/step - loss: 0.1294 - accuracy: 0.9507 - val_loss: 1.0420 - val_accuracy: 0.7354
Epoch 8/10
101/101 [==============================] - 11s 106ms/step - loss: 0.1147 - accuracy: 0.9616 - val_loss: 1.2843 - val_accuracy: 0.7540
Epoch 9/10
101/101 [==============================] - 11s 105ms/step - loss: 0.0930 - accuracy: 0.9671 - val_loss: 1.8285 - val_accuracy: 0.7217
Epoch 10/10
101/101 [==============================] - 11s 105ms/step - loss: 0.0855 - accuracy: 0.9699 - val_loss: 1.9735 - val_accuracy: 0.6969
Total time taken by the model: 2838.1581342220306 seconds
```

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

DenseNet model (with optimiser)

```
Epoch 1/10
101/101 [==============================] - 802s 8s/step - loss: 0.5626 - accuracy: 0.7616 - val_loss: 0.4111 - val_accuracy: 0.8037
Epoch 2/10
101/101 [==============================] - 11s 105ms/step - loss: 0.3744 - accuracy: 0.8292 - val_loss: 0.4154 - val_accuracy: 0.8273
Epoch 3/10
101/101 [==============================] - 10s 101ms/step - loss: 0.2885 - accuracy: 0.8723 - val_loss: 0.5264 - val_accuracy: 0.7925
Epoch 4/10
101/101 [==============================] - 10s 103ms/step - loss: 0.2233 - accuracy: 0.9086 - val_loss: 0.8865 - val_accuracy: 0.7627
Epoch 5/10
101/101 [==============================] - 10s 100ms/step - loss: 0.1777 - accuracy: 0.9275 - val_loss: 0.6794 - val_accuracy: 0.7590
Epoch 6/10
101/101 [==============================] - 10s 99ms/step - loss: 0.1455 - accuracy: 0.9473 - val_loss: 0.8133 - val_accuracy: 0.7379
Epoch 7/10
101/101 [==============================] - 10s 100ms/step - loss: 0.1155 - accuracy: 0.9551 - val_loss: 1.0163 - val_accuracy: 0.7578
Epoch 8/10
101/101 [==============================] - 10s 100ms/step - loss: 0.0818 - accuracy: 0.9709 - val_loss: 1.0706 - val_accuracy: 0.7839
Epoch 9/10
101/101 [==============================] - 11s 104ms/step - loss: 0.0768 - accuracy: 0.9715 - val_loss: 1.2101 - val_accuracy: 0.6832
Epoch 10/10
101/101 [==============================] - 10s 100ms/step - loss: 0.0690 - accuracy: 0.9771 - val_loss: 1.2956 - val_accuracy: 0.7031
Total time taken by the model: 912.7276697158813 seconds
```

Inception resnet (without optimizer)

```
Epoch 1/10
102/102 [==============================] - 1310s 13s/step - loss: 1.5030 - accuracy: 0.7010 - val_loss: 0.4473 - val_accuracy: 0.8463
Epoch 2/10
102/102 [==============================] - 22s 218ms/step - loss: 0.4986 - accuracy: 0.7891 - val_loss: 0.4059 - val_accuracy: 0.8315
Epoch 3/10
102/102 [==============================] - 22s 213ms/step - loss: 0.3588 - accuracy: 0.8476 - val_loss: 0.5111 - val_accuracy: 0.8265
Epoch 4/10
102/102 [==============================] - 22s 215ms/step - loss: 0.2808 - accuracy: 0.8878 - val_loss: 0.5334 - val_accuracy: 0.8030
Epoch 5/10
102/102 [==============================] - 23s 221ms/step - loss: 0.2053 - accuracy: 0.9246 - val_loss: 0.7353 - val_accuracy: 0.7819
Epoch 6/10
102/102 [==============================] - 24s 232ms/step - loss: 0.1765 - accuracy: 0.9419 - val_loss: 0.8133 - val_accuracy: 0.7807
Epoch 7/10
102/102 [==============================] - 23s 220ms/step - loss: 0.1416 - accuracy: 0.9539 - val_loss: 0.9409 - val_accuracy: 0.7943
Epoch 8/10
102/102 [==============================] - 23s 220ms/step - loss: 0.1268 - accuracy: 0.9576 - val_loss: 1.2106 - val_accuracy: 0.7794
Epoch 9/10
102/102 [==============================] - 23s 228ms/step - loss: 0.1100 - accuracy: 0.9666 - val_loss: 1.1047 - val_accuracy: 0.7968
Epoch 10/10
102/102 [==============================] - 23s 220ms/step - loss: 0.0955 - accuracy: 0.9706 - val_loss: 1.4298 - val_accuracy: 0.7770
Total time taken by the model: 1562.1777930259705 seconds
```

Inception resnet(with optimizer)

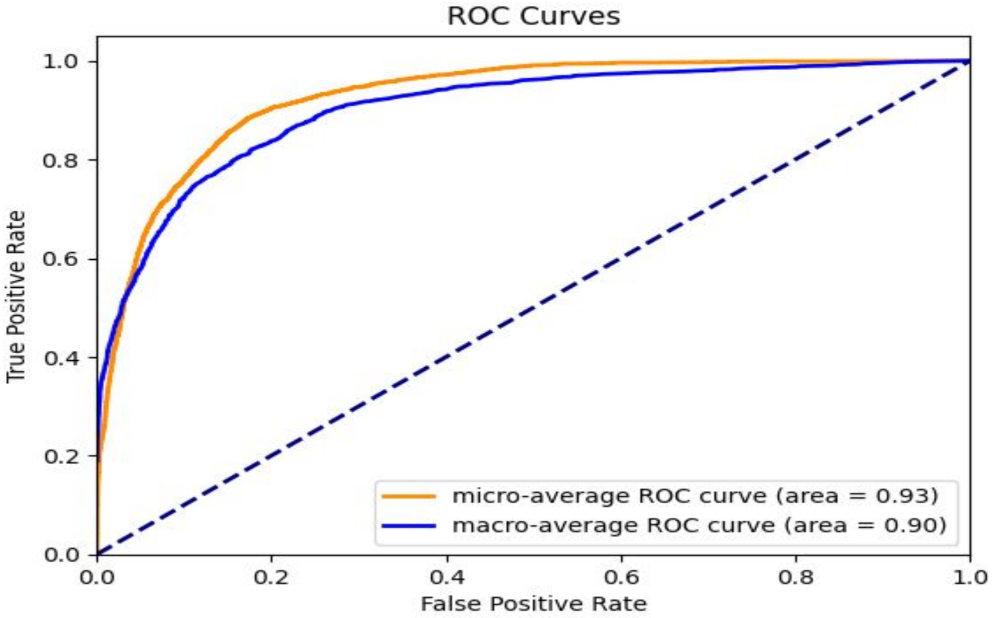
```
Epoch 1/10
102/102 [==============================] - 40s 254ms/step - loss: 1.0386 - accuracy: 0.7066 - val_loss: 0.5526 - val_accuracy: 0.7175
Epoch 2/10
102/102 [==============================] - 23s 221ms/step - loss: 0.4593 - accuracy: 0.7823 - val_loss: 0.4629 - val_accuracy: 0.8228
Epoch 3/10
102/102 [==============================] - 22s 216ms/step - loss: 0.3490 - accuracy: 0.8550 - val_loss: 0.6563 - val_accuracy: 0.7559
Epoch 4/10
102/102 [==============================] - 23s 225ms/step - loss: 0.2590 - accuracy: 0.8949 - val_loss: 0.5849 - val_accuracy: 0.7745
Epoch 5/10
102/102 [==============================] - 24s 235ms/step - loss: 0.2116 - accuracy: 0.9184 - val_loss: 0.6715 - val_accuracy: 0.7770
Epoch 6/10
102/102 [==============================] - 22s 217ms/step - loss: 0.2261 - accuracy: 0.9159 - val_loss: 0.7235 - val_accuracy: 0.7869
Epoch 7/10
102/102 [==============================] - 23s 222ms/step - loss: 0.1497 - accuracy: 0.9437 - val_loss: 0.8575 - val_accuracy: 0.7596
Epoch 8/10
102/102 [==============================] - 23s 227ms/step - loss: 0.1302 - accuracy: 0.9601 - val_loss: 1.0228 - val_accuracy: 0.7633
Epoch 9/10
102/102 [==============================] - 23s 220ms/step - loss: 0.0814 - accuracy: 0.9722 - val_loss: 1.0612 - val_accuracy: 0.7497
Epoch 10/10
102/102 [==============================] - 23s 222ms/step - loss: 0.1027 - accuracy: 0.9647 - val_loss: 1.1000 - val_accuracy: 0.7869
Total time taken by the model: 245.01757049560547 seconds
```

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

ResNet50 model metrics

	Without Optimizer	**With Optimizer(adam)**
Accuracy	0.764	**0.801**
Precision Score	0.900	**0.899**
Recall Score	0.856	**0.871**
F1 Score	0.865	**0.876**
Error Rate	0.235	**0.218**
Computational Time(in sec)	8121.742	**6101.651**

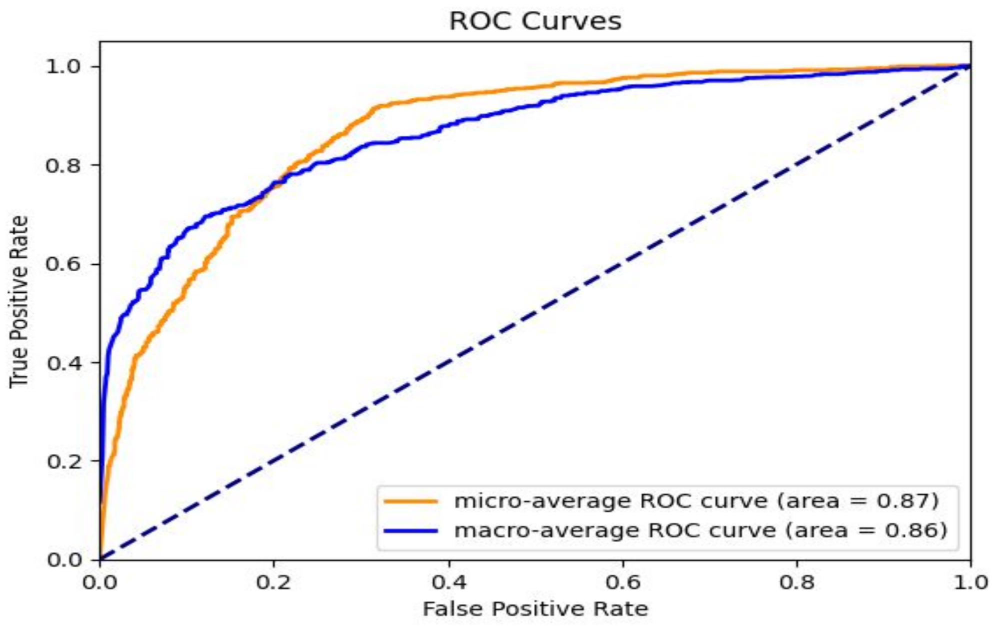
ROC Curve and AUC(ResNet50)



DenseNet model metrics

	Without Optimizer	**With Optimizer(adam)**
Accuracy	0.699	**0.720**
Precision Score	0.718	**0.871**
Recall Score	0.776	**0.843**
F1 Score	0.820	**0.836**
Error Rate	0.308	**0.300**
Computational Time(in secs)	2838.158	**912.727 seconds**

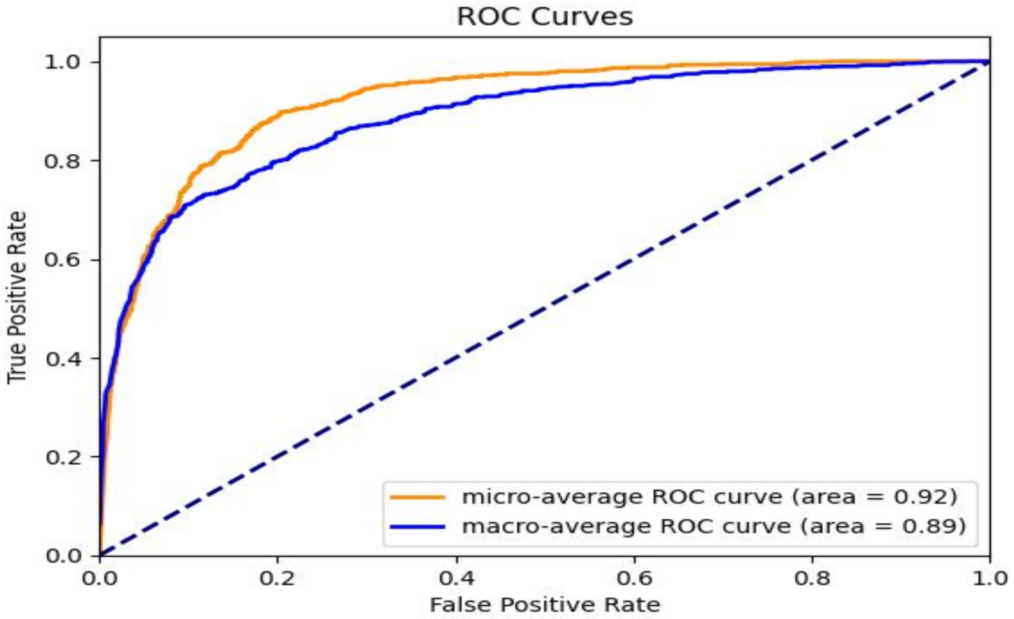
ROC Curve and AUC(DenseNet)



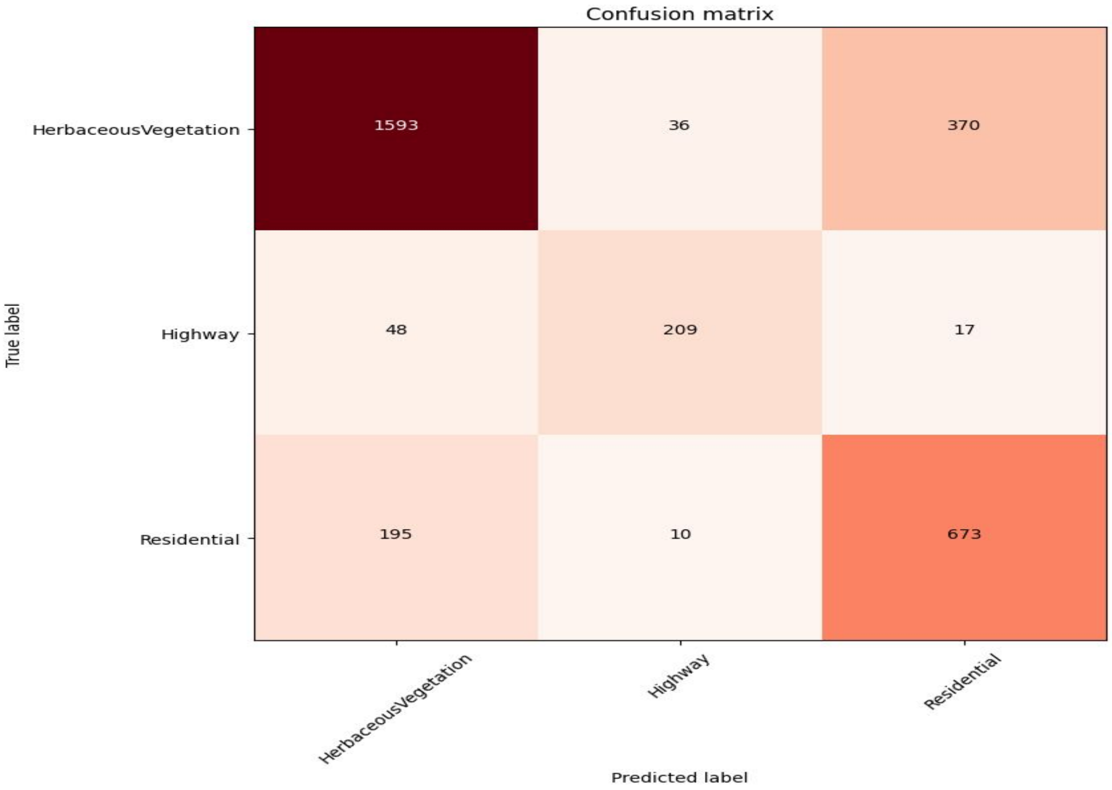
Inception resnet model metrics

	Without Optimizer	**With Optimizer(adam)**
Accuracy	0.777	**0.780**
Precision Score	0.842	**0.877**
Recall Score	0.828	**0.895**
F1 Score	0.865	**0.872**
Error Rate	0.277	**0.223**
Computational Time(in sec)	1562.177	**245.017**

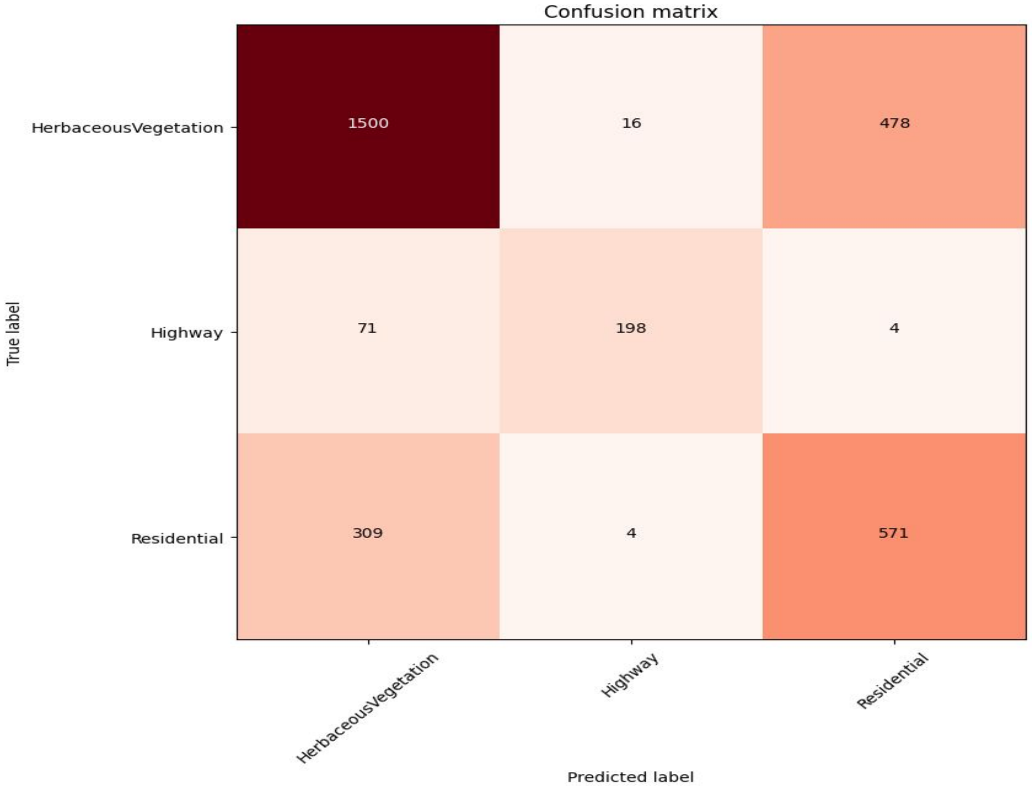
ROC Curve and AUC(Inception resnet)



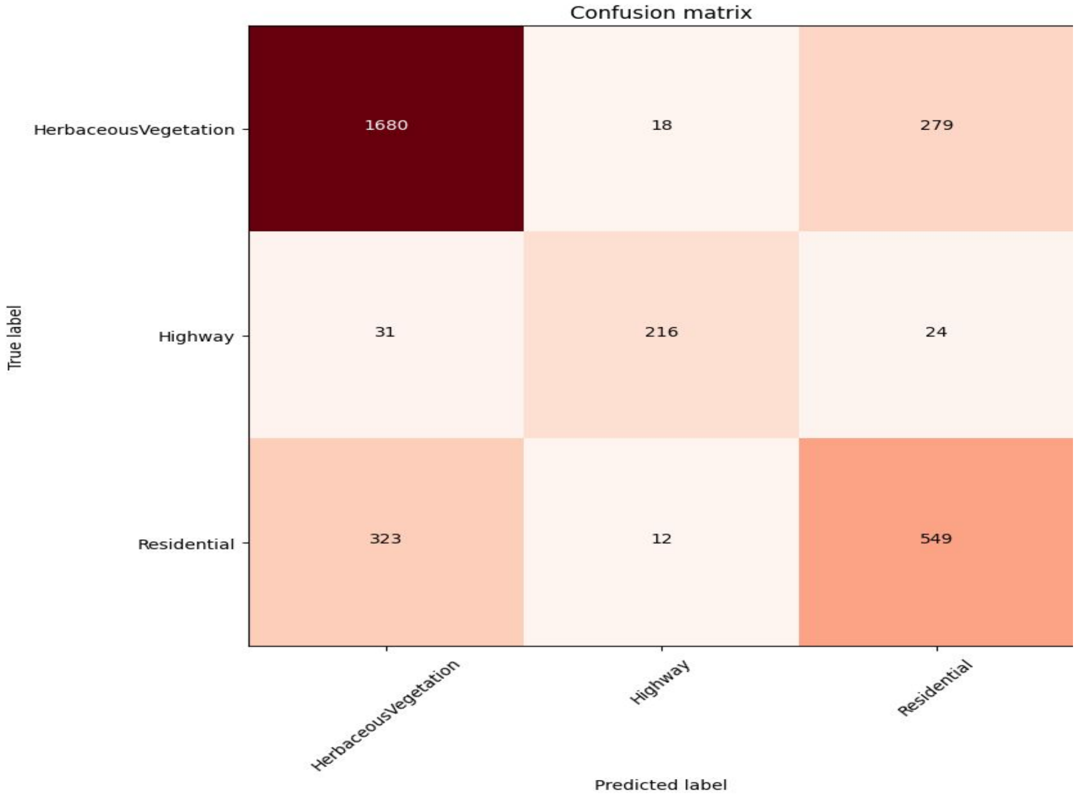
Confusion matrix (ResNet50)



Confusion matrix (DenseNet)



Confusion matrix (Inception)



RESULT ANALYSIS

- The accuracies of 4 different clustering methods(K-Means, Mean Shift, DT-CWT, PSO) are compared using a table consisting of Pfa, Pma Pte as comparison metrics.
- In image clustering, the clustering method with least metric value is considered the most effective.
- In image classification, the input dataset is fed into a deep learning algorithm(ResNet model) for better effective and efficient classification.
- The accuracy at each epoch is varied and the final epoch's(10) accuracy of 81% is achieved.
- The accuracy follows an increasing trend when plotted against epoch.
- The loss over each epoch is significantly decreasing with increase in epoch.
- In the confusion matrix, the true positives(diagonal elements) are high implying the overall accuracy is maximum.

CONCLUSIONS

- From the histogram the PSO method has the least Pfa, Pte and Pma values, implying that it is the most effective method.
- In image classification, the accuracy is optimal considering the dataset size and number of parameters.
- The final change mask can also be used for image segmentation in order to quantify the individual changes.
- The classification output can be used in calculating land cover land change over the years.

Workflow

0-2 weeks	Literature Survey
2-3 weeks	Data Collection
3-5 weeks	Algorithm Formation
5-9 weeks	Algorithm Implementation
9-10 weeks	Documentation

References

- Base Paper
- https://www.hindawi.com/journals/complexity/2020/6693411/
- LULC using deep learning
- Particle Swarm Optimization

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA
THANJAVUR | KUMBAKONAM | CHENNAI

THANK YOU