Table of Content

Abstract:

The goal of this study is to create a cutting-edge computer vision-based meteor detection system that makes use of artificial intelligence and machine learning. The main goal is to precisely identify meteors in sky photographs and evaluate three cutting-edge meteor detection methods. The paper explores several tactics, such as data augmentation and the choice of suitable classification models for training on restricted data, to address the difficulties given by managing small and skewed datasets. The most efficient method for meteor detection will be determined by comparing these models. The goal of the project is to improve our comprehension of meteors and their interactions with the atmosphere of Earth through the multidisciplinary fusion of astronomy, computer vision, and machine learning. If this work is successful, it will not only improve the ability of the scientific community to explore extraterrestrial occurrences but also expand computer vision applications beyond meteor detection.

Declaration Statement:

I hereby affirm that the work included in the research article titled "Discovering Meteors from Images Using ML" is entirely original to me. Each and every source used for this research has been properly acknowledged and credited. Any contributions from outside sources have been duly acknowledged. The study adheres to the moral principles and academic standards that have been established by the institution and the scientific community.

I further attest that all contributors' intellectual contributions, including those of the advisors, have been properly acknowledged. No portion of this research has ever been submitted for a grade or other academic evaluation.

I acknowledge that this work is free from any academic dishonesty or plagiarism because I am aware of the negative effects of doing so. To the best of my knowledge, the information and findings in this study are true and trustworthy.

I understand that breaking any of the disclosures could result in serious academic repercussions, and I am prepared to assist with any inquiries into the veracity and originality of this research.

______________________ is my signature.

Date: __________________________

1. Introduction:

Both scientists and stargazers are captivated by meteors, those dazzling streaks of light in the night sky. In addition to providing a spectacular celestial display, they are essential for comprehending the intricate interactions between these celestial guests and our planet's environment. Their research sheds light on the interactions of different atmospheric constituents as well as severe ablation phenomena and hypervelocity physics. As technology and AI develop, a fascinating new path opens in the form of computer vision, a potent tool that has the potential to fundamentally alter meteor detection and classification.

Radio frequency analysis has historically been used to detect meteors, but current advances in computer vision offer a promising alternative method. Using cutting-edge computer vision techniques, this project attempts to recognize and categories meteors from a wealth of sky photos provided by the renowned Bayfordbury Observatory. The.FIT files that correlate to these photographs, which were recorded as JPG files, allow for a thorough investigation of the meteor outbursts.

Some of the most relevant methods used for meteors detection are as follows:

When meteoroids fragment in the Earth's atmosphere, bright streaks known as meteors are visible in the sky. A practical way to learn more about the distribution of minerals in our solar system is by studying meteors. Radar, passive radio detectors, all-sky photography, CCD (charge-coupled device) cameras, digital video cameras, and TV cameras with optional image intensifiers are only a few of the techniques used to observe meteors. While camera-based methods must be used at night, radio-based detection can be done during the day. However, camera-based observations allow the creation of a light curve that illustrates the meteor's time dependent fluctuations in light output and provides important details about the mass and structure of the original particle or parent object, such as comets and asteroids. The measurement of the meteoroid's chemical composition is also possible with wide-band observation using well-constructed photometric filters. While camera-based systems are more frequently employed, certain approaches combine several observation techniques. The best allsky cameras for seeing powerful light events like bolides and fireballs have great spatial resolution and lengthy exposure durations. On the other hand, determining a meteor's air route necessitates concomitant measurements from at least two separate sites, preferably with high temporal resolution. Additionally, a faster frame rate in camera recordings offers more information for precise modelling of the meteoroid's structure [1].

ConvNet, commonly known as the Convolutional Neural Network (CNN), is a deep feed-forward architecture that is renowned for its outstanding capacity to generalize more successfully than networks with completely connected layers. In a biologically inspired approach, CNN is motivated by the idea of hierarchical feature detectors, allowing it to learn extremely abstract features and effectively recognize things. Its advantage to classical models is explained by several important factors. First, CNN uses the idea of weight sharing to drastically cut down on the number of parameters that need to be learned, which enhances generalisation. The fewer parameters make training easier and limit the chance of overfitting. The feature extraction and classification phases are combined into a single learning process by

CNN. The network's capacity to learn pertinent features and create precise classifications is improved by this integration. Thirdly, as compared to general models of artificial neural networks (ANN), CNNs are better at implementing huge networks. CNN is more adaptable and scalable because complicated structure design and implementation are more controllable in CNN. CNNs have a wide range of uses because of their exceptional performance, including image classification, object detection, face detection, voice recognition, vehicle recognition, diagnosis of diabetic retinopathy, facial expression recognition, and many more. The purpose of this study is to create a theoretical framework that improves our knowledge and comprehension of CNNs [2].

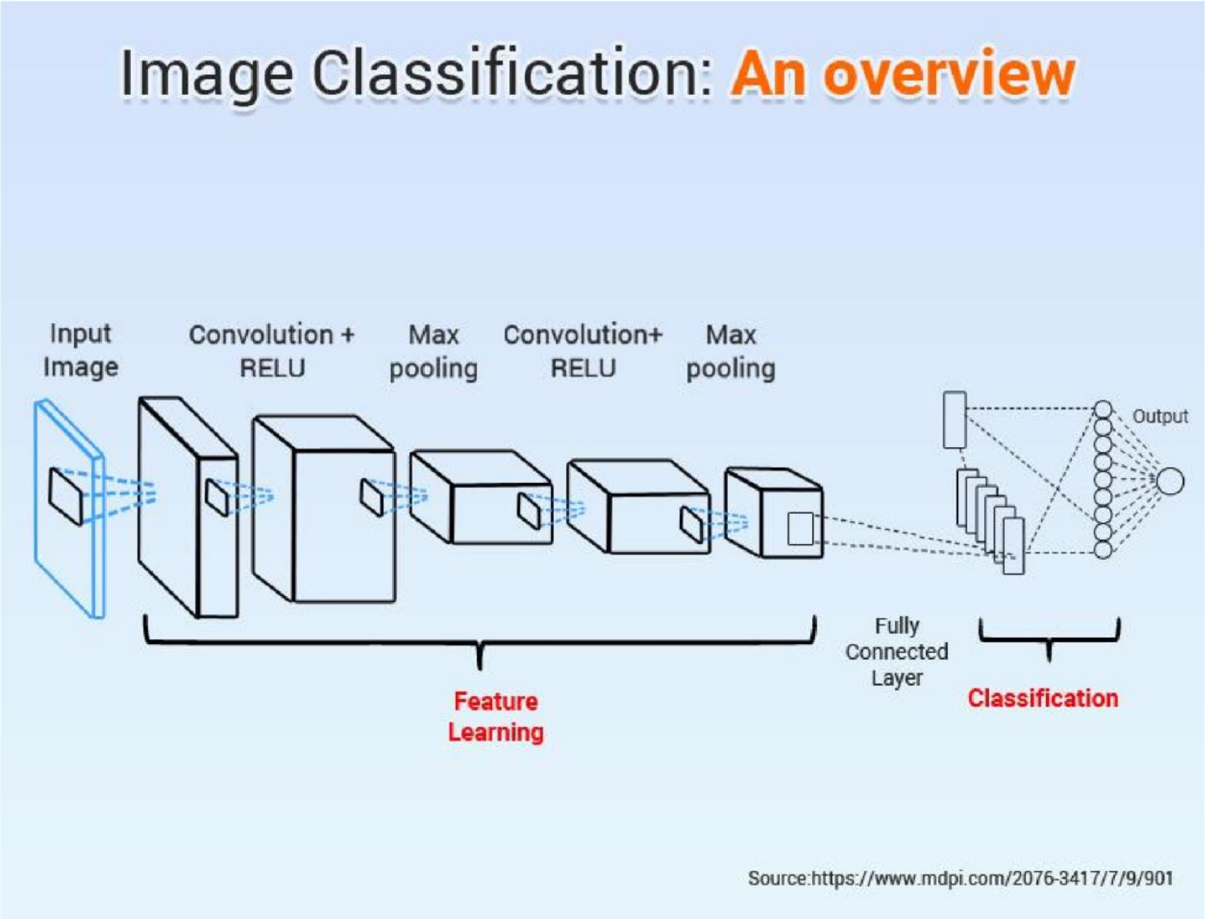


Fig1: CNN Image Classification (Web Source: https://www.aismartz.com/blog/imageclassification-an-overview/)

Optical observations are essential for meteor observing, but it is difficult to automate their detection since MetRec, MeteorScan, and UFOCapture—the current detection pipelines—need human intervention to filter out false positives. Convolutional Neural Networks (CNNs) have successfully been implemented in projects like the CAMS network and EXOSS, demonstrating great proficiency in filtering out false positive images from meteor sightings. In this study, we provide a tailored pipeline with three major stages: data preprocessing, CNN filtering and modeling, and evaluation [3].

The main goal of this research is to use cutting-edge computer vision algorithms to accurately identify meteors in the provided sky photos. I want to build a reliable and accurate meteor detection system that can classify these celestial phenomena by utilizing the capabilities of machine learning and artificial intelligence. The study offers the chance to compare and assess three cutting-edge models as part of a thorough analysis of meteor detection within the context of machine learning. The benefits and drawbacks of each model will be carefully evaluated through this comparison study, which will help determine the best course of action for this application. Handling tiny, skewed datasets is another key topic covered in this research. To overcome this restriction, several strategies have been investigated, including methods like data augmentation and the careful selection of classification models suitable for training on small amounts of data. These initiatives aim to increase the meteor detecting system's overall robustness and effectiveness. I set out on a fascinating adventure to deepen our understanding of meteors and how they interact with the Earth's atmosphere through this fusion of astronomy, computer vision, and machine learning. I seek to improve the scientific community's ability to solve the riddles of these extraterrestrial visitors by embracing the most recent technological developments. A successful outcome will also make a significant contribution to the field of computer vision, opening new opportunities for uses beyond meteor detection, furthering our understanding of science.

2. Literature Review:

For processing and analyzing visual input, Convolutional Neural Networks (CNNs) have become a potent family of deep learning models. By exhibiting astounding performance in tasks like picture classification, object identification, image segmentation, and more, they have revolutionized a variety of industries, particularly computer vision. CNNs are particularly well suited for image-based applications because they draw inspiration from how the biological visual system works to acquire hierarchical characteristics from raw data. The capacity of CNNs to automatically learn spatial feature hierarchies using convolutional and pooling layers is one of their key qualities. By convolving across the input data, the convolutional layers use filters and kernels to accomplish feature extraction, identifying patterns like edges, textures, and forms. The spatial dimensions are then reduced by pooling layers, keeping only the most important data. CNNs are resistant to translation, rotation, and partial occlusion in pictures thanks to this hierarchical feature learning.

In image classification tasks, CNNs have demonstrated their efficacy, achieving astounding accuracy on benchmark datasets like ImageNet. As an illustration, the VGG-16 model, which Simonyan and Zisserman first presented in 2014 [4], performed exceptionally well in the ImageNet Large-Scale Visual Recognition Challenge. Following this achievement, a number of enhanced architectures have been put forth, each pushing the limits of picture classification accuracy, including ResNet by He et al. [5], InceptionV3 by Szegedy et al. [6], and EfficientNet by Tan and Le [7]. Another important area where CNNs have significantly contributed is object detection. Real-time object detection has been made possible by models like YOLO (You Only Look Once), developed by Redmon et al. [9] and Faster R-CNN, introduced by Ren et al. [8], and by Ren et al. To locate and identify objects in an image, these models effectively integrate CNNs with region proposal processes or anchor-based methods.

CNNs have been used for semantic segmentation in addition to picture classification and object detection. An innovation in this field was the Fully Convolutional Networks (FCNs) introduced by Long et al. [10]. To forecast pixel-wise segmentation masks, FCNs use transposed convolutions. This allows them to be useful in tasks like medical image analysis, autonomous navigation, and video comprehension. CNNs have discovered uses in a variety of fields beyond of conventional computer vision jobs. CNNs have had a substantial impact on both natural language processing (NLP) and natural language understanding (NLU), particularly in text categorization problems. By utilising one-dimensional convolutions over word embeddings, Kim's work on "Convolutional Neural Networks for Sentence Classification" [11] showed the potential of CNNs in NLP problems. Additionally, CNNs have been effectively used in transfer learning, which involves modifying previously learned models for a given task using big datasets. By enabling the construction of applications with minimal labelled data and processing resources, this transfer learning paradigm has expanded the applicability of CNNs to a variety of real-world contexts.

Convolutional Neural Networks (CNNs) have emerged as a game-changing technology in computer vision and other fields. Significant improvements in image classification, object detection, semantic segmentation, and even natural language processing tasks have been made possible by their hierarchical feature learning, sophisticated architectures, and transfer learning capabilities. In the upcoming years, it is anticipated that CNNs will continue to be at the forefront of cutting-edge applications as deep learning research advances.

When meteoroids fragment in the Earth's atmosphere, light streaks known as meteors are visible in the sky. A practical technique to comprehend the distribution of matter in our solar system is through meteor observation. Radar, passive radio detectors, all-sky photography, CCD (charge coupled device) cameras, digital video cameras, or television (TV) cameras with an optional picture intensifier are frequently used for meteor observations. While camera-based approaches can only be used at night, radio-based detection techniques can be used during the day, making them suited for estimating the total amount of meteor activity. Despite this drawback, camera-based studies enable the construction of the light curve, or the time-dependent fluctuations in a meteor's light, which may reveal details about the mass and composition of the originating particle or parent object, such as comets and asteroids. A well-constructed bank of photometric filters and wide-band observation both enable learning more about the meteoroid's chemical make-up. Although camera-based systems are more prevalent, it is also possible to combine several types of observations. The best all-sky cameras for seeing powerful light phenomena, such as bolides or fireballs, are those with a high spatial resolution and lengthy exposure times. However, it is necessary to view meteors concurrently from at least two different locations, optionally with high temporal resolution, to calculate the atmospheric track. Additionally, additional data is available for the modelling of the meteoroid structure at a greater frame rate. To estimate meteoroid trajectories, meteor observation using two or more camera systems has become the norm. The Spanish Meteor Network (SPMN) has about 25 video and CCD stations; Cameras for All Sky Meteor Surveillance (CAMS) operates more than 60 narrow-field cameras at five locations across the United States (three in California, one in Florida, and on the Mid-Atlantic coast). There are networks with varying sizes and technological capabilities. Later, amateur astronomers in the Netherlands and Belgium used the idea. More than 30 cameras are used by the Croatian Meteor Network (CMN). The 36 constantly operating stations that make up the Polish Fireball Network (PFN) have 57 sensitive analogue video cameras and 7 high-resolution digital cameras. Four CCD cameras connected to 18-mm image intensifiers are used by the Canadian Automated Meteor Observatory (CAMO) and operate at a frame rate of 80. The Desert Fireball Network (DFA) currently uses 49 digital single reflex camera (DSLR)-based stations with a nominal spacing of 130 km to cover one third of Australia (about 2.5 million km2). All of France is covered by the Fireball Recovery and Interplanetary Observation Network (FRIPON), which has 100 allsky cameras and an average station distance of 100 km. The Italian network PRISMA (Prima Rete per la Sorveglianza sistematica di Meteore e Atmosfera), which aims to use gamma-ray spectrometers to reveal the existence of short-lived cosmogenic radioisotopes, is also being developed in collaboration with the FRIPON. [12]

The field of artificial intelligence (AI) known as computer vision (CV) has gained prominence and is now used widely in a variety of fields, including astronomy, security, automobile, and night sky monitoring. To recognise items in visual media, including both static and moving images, CV mostly uses pattern recognition algorithms. The most popular strategy in CV now is supervised AI, which trains on enormous datasets of images with human labels. Computer vision's application of machine learning (ML) and deep learning (DL) models has advanced significantly, particularly when it comes to tasks like object localisation and recognition. Astronomical image analysis stands out as a field of particular interest for researchers among the practical applications. The lack of appropriate annotated data sets, particularly in terms of resolution and scale, poses a serious problem for this field. As a result, solving this problem requires a vast amount of storage space as well as powerful computing resources. We examine the cutting-edge approaches that have been created over the past ten years and provide a comprehensive evaluation and analysis of the various issues that astronomers face in this field. [13]

3. Methodology:

This chapter discusses architecture and methods used to perform meteor classification in sky images. The project was implemented on Google Colaboratory on Python through multiple libraries such as numpy, pandas, matplotlib, scikit, pytorch etc. It spanned multiple stages starting from data collection, storage, preprocessing, augmentation, and finally training and visualization. Data was categorized into two classes, planes and meteors and assorted into folders for train, validation, test split. After preprocessing data, details of which are shared below, three state of the art classification algorithms, Resnet50, Googlenet, Inception were trained, and their results were compared. Models were hyper tuned and their performance was then evaluated on unseen test data. The figure below highlights integral steps that were part of this project.

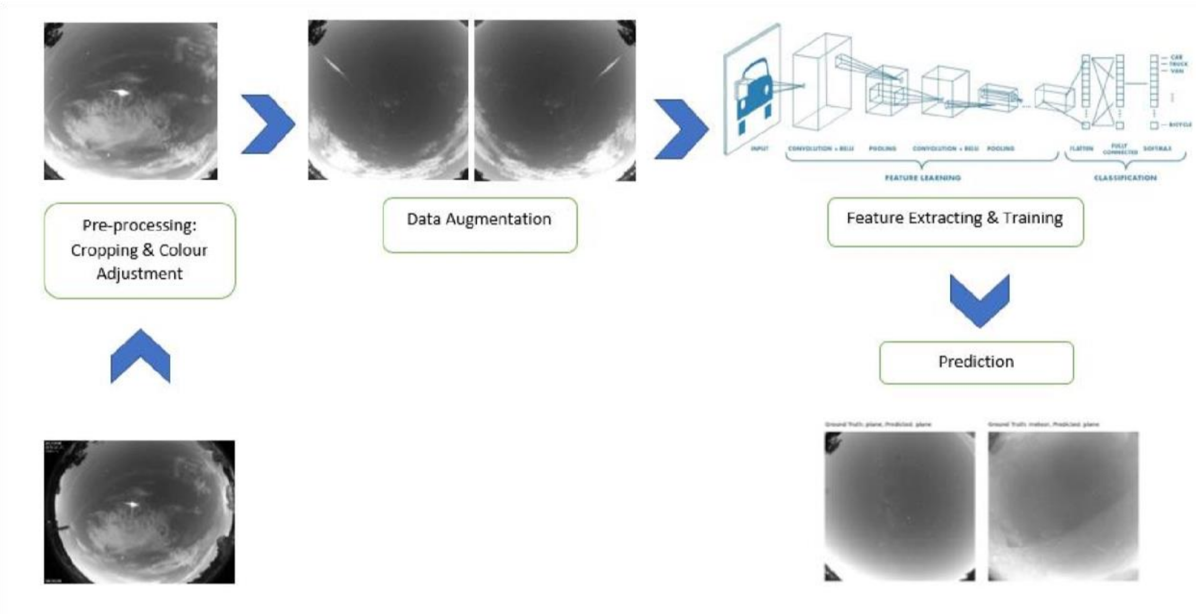


Fig 3.1: Pipeline comprising of integral steps performed in this work.

3.1 Dataset:

Data consisted of sky images obtained from Bayfordbury Observatory and had 158 images in total. 118 images were plane images whereas 40 images had meteor trails in them. Each image had dimensions of 640x480, and had either of two labels, plane, or meteor. Figure shows a sample of images from the dataset.

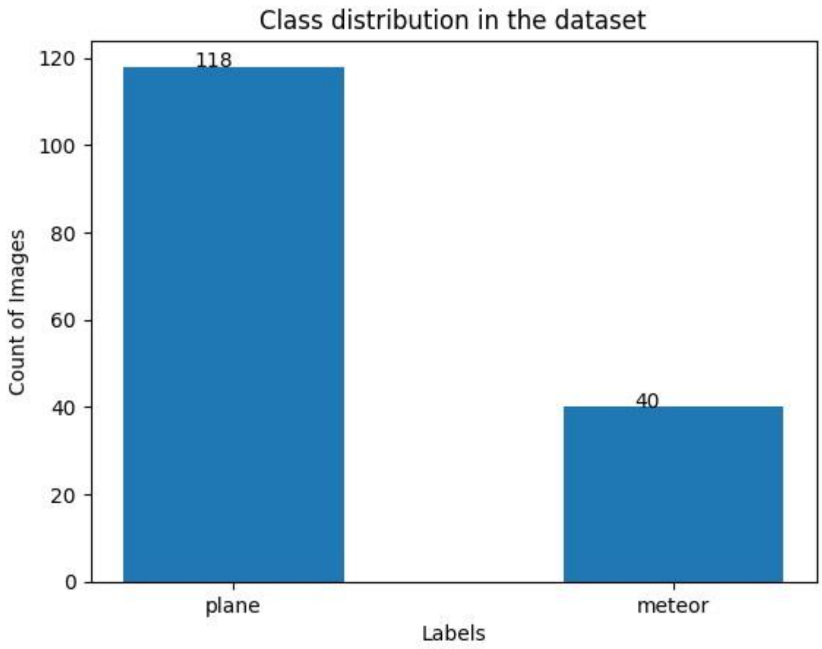


Fig 3.2: Number of images belonging to each class

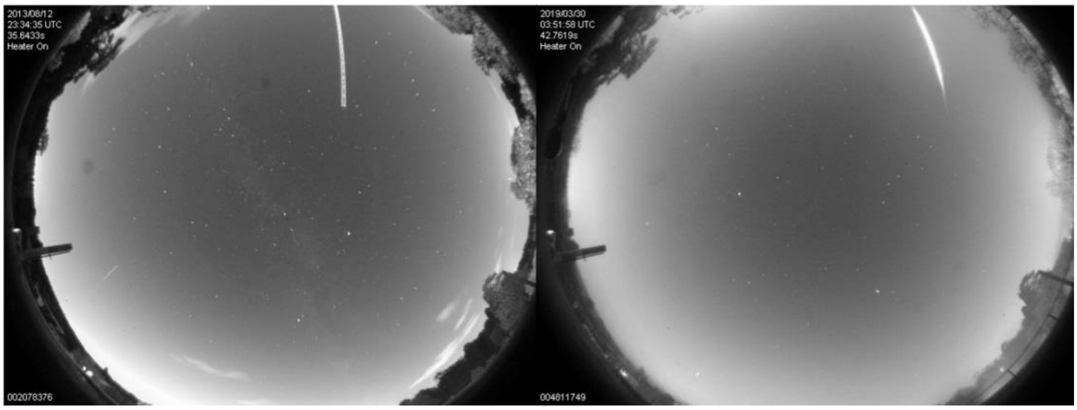


Fig 3.3: Raw data: Left image: Plane, Right image: Meteor

3.2 Preprocessing

Raw images obtained from the observatory had to be processed to perform deep learning on them. The sky images had to be cropped as they had irrelevant data on the borders of each image, as shown in fig, as this could have potentially added bias in the model training. Through Pytorch's transformation images were center cropped with size 450. This was done to ensure minimum noise was incorporated and images only had useful shots of planes/meteors. All these images were then visually inspected to make sure that there was no significant data loss in the process.

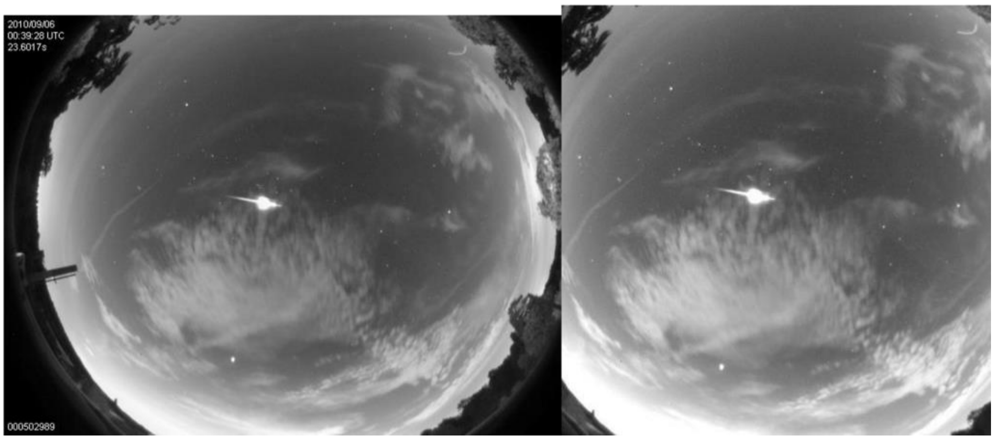
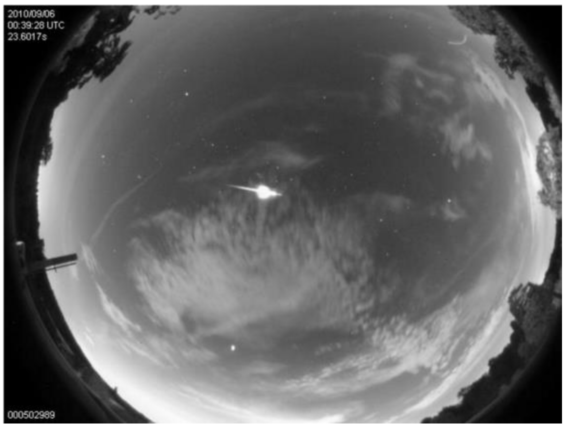


Fig 3.4: Preprocessing: Left: Raw uncropped image, Right: Center Cropped image of size 450

In the second stage of preprocessing, images had their brightness level adjusted as originally, they were dark, and streaks of meteors were not visible in them. Pytorch's adjust_brightness functional transformation was used and experimented multiple times to pick the ideal value of the brightness factor. 1.2 appeared to be an ideal brightness factor. Values greater than 1.2 were resulting in increased whiteness in the images, and lesser values did not add significant brightness. Fig below difference in brightness at three different levels



Original Image

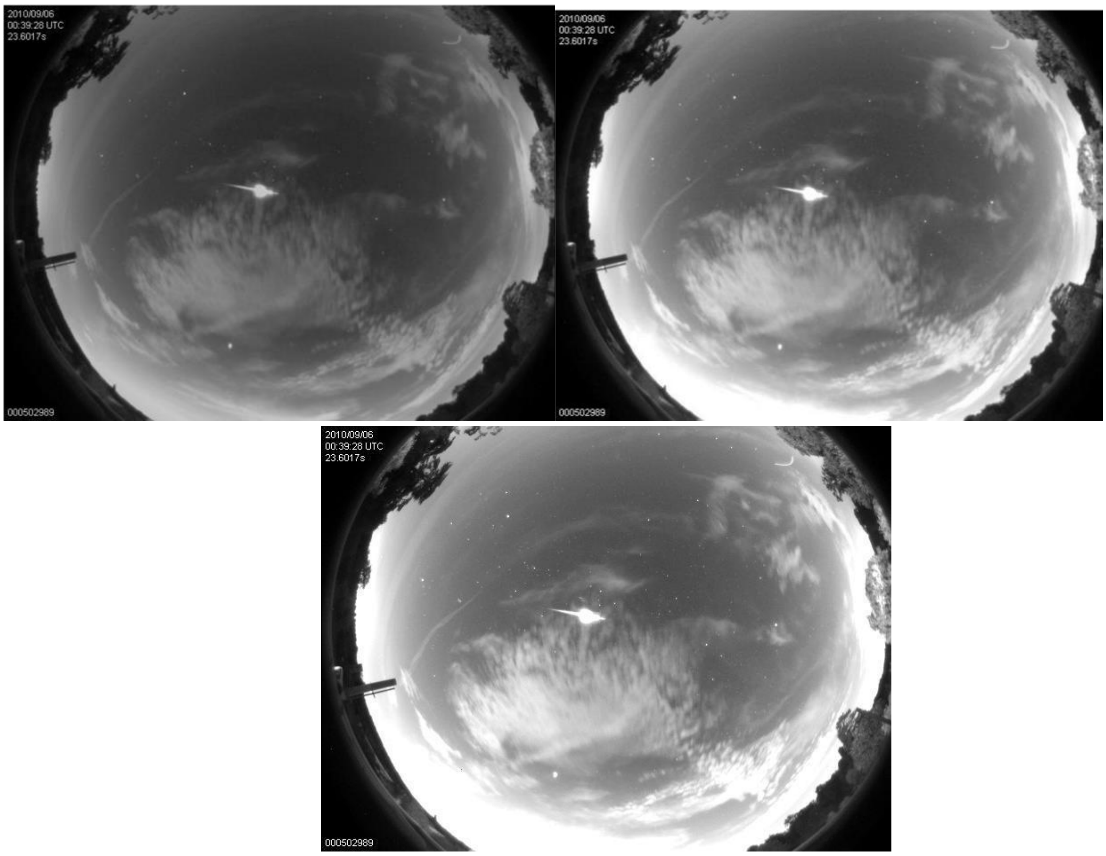


Fig 3.5, In clockwise: Image at 0.8 level brightness, at 1.2 and third image at 1.4 factor of brightness.

3.3 Data Augmentation

It is a well-known understanding that deep learning models are data hungry, as they not only need volume in data but also variety within the dataset.1

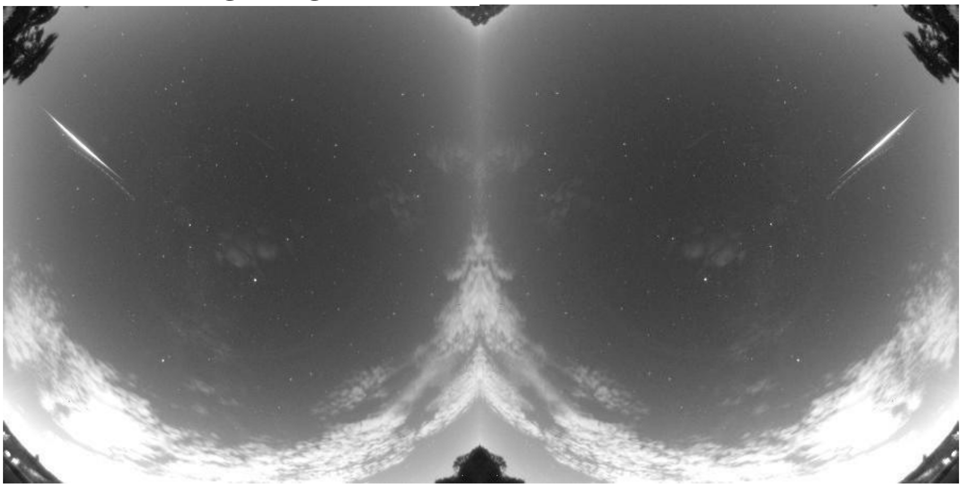
Data augmentation techniques are widely used in deep learning projects to artificially add new data points in the dataset that help to increase model robustness. To increase generalizability of the model it is imperative that the model is trained on real world examples, but unfortunately acquiring such large-scale data is often not cost effective. This is where data augmentation techniques come to assistance as they add representation to the data through various mathematical transformation techniques.

Therefore, to bring variety and volume to the dataset, both online and offline data augmentation methods were employed in this project.

3.3.1 Offline Augmentation:

It is very common for deep learning models to get trained on thousands of images. But in this project, one of the challenges was inadequate data quantity. Total number of images used were 158 which is significantly lower if compared to general training practice. In such cases, geometric transformations 2 are commonly used to not only increase the number of samples but also to balance the classes, as images belonging to planes were roughly thrice the size of meteor images. Horizontal flip was applied to randomly selected subset of training images, but not to validation or test images; this added 46 new images to the plane category and 20 images to the meteor set making the total count 130 and 49 for both categories respectively (fig 5)

One of the concerns when applying data augmentation is that excessive augmentation techniques result in data bias, that is, augmented data ends up being different from original data distribution and adds noise as the result 3. To avoid this, it was made sure that the number of new synthesized images is less than the original count and not a lot of techniques are applied that would disturb original data distribution. Fig below shows the result of horizontal flip to one of the trainings images.



Original meteor image Augmented image after horizontal flip

Fig 3.6: Before and after data augmentation

3.3.2 Online Augmentation:

The augmentation type that is used on a batch of images prior to training is called online or on-the-fly training. During online augmentation, random transformations are applied to images from the original set and then sent to the training model. Unlike offline augmentation where augmented images are stored in memory, images are not stored anywhere, and it is not possible to predict which images are augmented. This type of augmentation further aids in adding variance to the data making it more generalized and robust. The following code snippet shows transformations applied to the training set randomly. Along with vertical flip contrast was introduced to add sharpness to the images. The images were then converted to tensort and normalized according to Imagenet standards.

```
image_transforms = {
    # Train uses data augmentation
    'train':
    transforms.Compose([
        transforms.Resize(size=256),
        transforms.RandomVerticalFlip(),
        transforms.ColorJitter(contrast=0.3),
        transforms.ToTensor(),
        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))  # Imagenet standards
    ]),
    'validation':
    transforms.Compose([
        transforms.Resize(size=256),
        transforms.ToTensor(),
        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
    ]),

    'test':
    transforms.Compose([
        transforms.Resize(size=256),
        transforms.ToTensor(),
        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
    ]),
}
```

Fig 3.7: Snippet of Pytorch code showing implementation of online augmentation.

Classification

CNNs are deep learning architectures which are stacked with different kinds of layers, such as convolution layer, max pooling layer from which extracted features are passed onto fully connected layers to generate required output. CNNs can learn visual features from images which are pertinent to different labels and are thus able to map that information to sophisticated high level understanding of domain knowledge.

In this work, three different kinds of architecture were explored. While researching these architecture, size of the dataset was kept in consideration. Complex models tend to over fit on a small set of images, therefore I selected those models that were lightweight and purposely built to cater to the issue of over fitting.

Resnet 50:

Residual Network or ResNet have been proposed in 2015 by Microsoft to overcome the issue of vanishing gradient when adding more layers to the network [17]. Adding more layers to a model helps in learning complex features but also diminishes gradients of the networks increasingly. ResNet tackles this problem by adding skips connection (Fig 3.6).

These skip connections allow alternate pathways for gradients to flow and learn the underlying mapping between the layers. Resnet has many variants, but in this project, ResNet 50 was used which has 50 layers in its network.

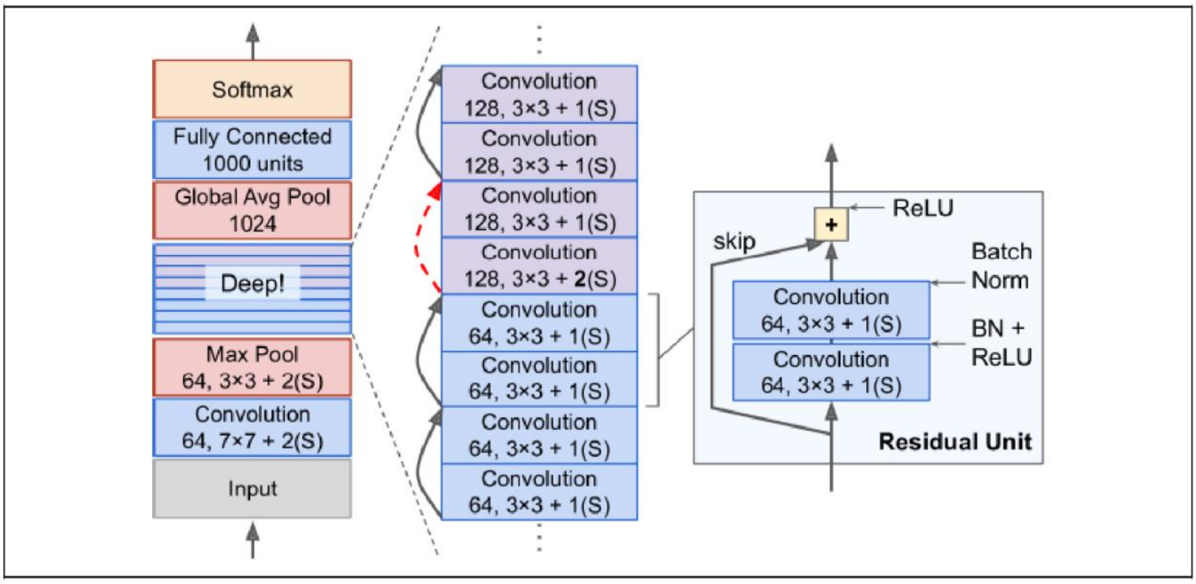


Fig 3.8 Skip connections are an integral part of ResNet. Inputs can flow through either. through weight layer or through identity link making this network more dynamic to learning. (Web Source: https://jason-adam.github.io/resnet50/)

ResNets also use bottleneck residual blocks to reduce the number of parameters in the network. Bottleneck residual blocks are made up of a stack of convolutional layers, followed by a 1x1 convolution layer that reduces the dimensionality of the feature maps. This helps to improve the efficiency of the network, while still preserving the accuracy.

ResNet have proved to be quite fast and accurate for variety of tasks, including image classification, object detection and semantic segmentation, achieving top-5 error rates of around 2% on the ImageNet dataset.[20]

ResNets have been very successful in a variety of tasks, including image classification, object detection, and semantic segmentation. They are a powerful tool for computer vision, and they are likely to continue to be used in many applications in the future.

GoogLeNet

GoogLeNet was developed by researchers at Google in 2014 to solve image classification tasks. It has 22 parametrized layers which are convolutional Layers and fully connected layers, and 5 non parameterized layers like Max-Pooling. GoogLeNet is built on Inception Net and therefore utilises Inception modules, which allow the network to choose between multiple convolutional filter sizes in each block.

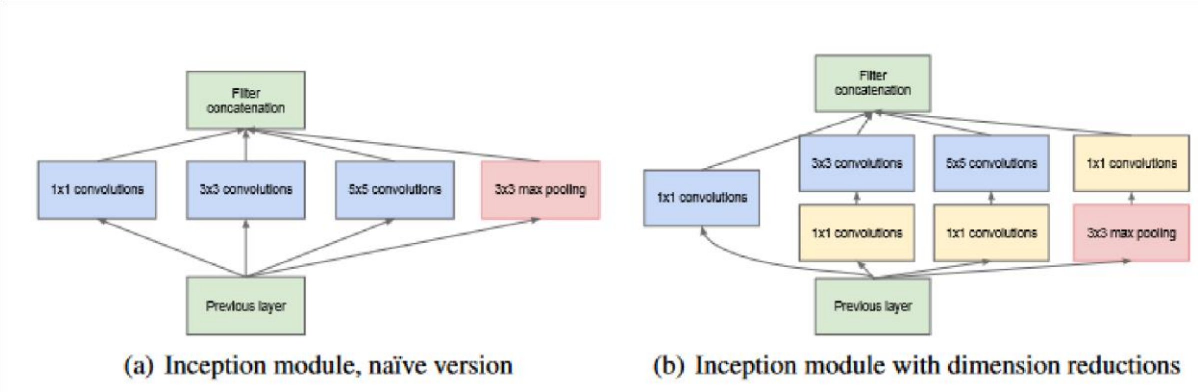


Fig 3.9: Difference between inception modules

One of the key motivations behind developing GoogLeNet, as mentioned by the researchers [8], was to develop a deep and better-performing network without increasing the number of parameters at each level, as enlarged architectures give rise to overfitting of the model and demands expensive computational resources. GoogLeNet was thus created keeping all these constraints in consideration.

Its salient features include inception modules (Fig 3.9) with dimensionality reduction. The naive approach of inception modules involves a large set of operations after convolving with a 5x5 filter. GoogLeNet reduced computational complexity and number of parameters by applying a different set of convolutions filters at multiple scales.

Furthermore, adding varying sizes of filters in convolution layers allow the network to extract features at different scales and thus maximise learning from an image [21]

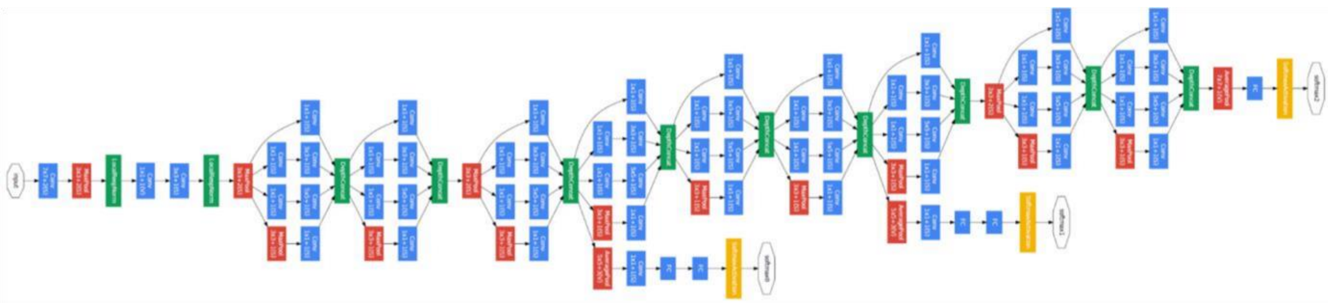


Fig 3.11: Architectural details of GoogLeNet

GoogLeNet showed promising results achieving accuracy, with a top-5 error rate of 6.67% on the ILSVRC 2014 dataset. It is also relatively fast, with an inference time of 0.9 seconds per image on a GPU. [21]

MobileNetV3

A convolutional neural network design called MobileNetV3 is renowned for its exceptional qualities and adaptability, especially in situations involving short datasets and constrained processing resources. One of MobileNetV3's primary features is its focus on efficiency and lightweight design, which makes it the perfect option for situations with limited resources, such mobile devices, and embedded systems. The hard swish activation function and a brand-new inverted residual structure with linear bottlenecks are just two of the network's original features. These characteristics considerably lessen the computational workload without sacrificing the performance and accuracy of the model. Additionally, Neural Architecture Search (NAS), a technique introduced by MobileNetV3, automatically optimizes the architecture for certain workloads, resulting in increased efficiency and adaptability. Researchers and developers may successfully complete tasks with this network even with little training data, making it an invaluable asset for a variety of real-world applications. For instance, MobileNetV3 can still produce results that are comparable with larger models in picture classification applications where the dataset size may be constrained. These benefits have been shown in research studies, like the work by Howard et al., where MobileNetV3's effectiveness on small datasets is highlighted, reiterating its practical utility and relevance [22].

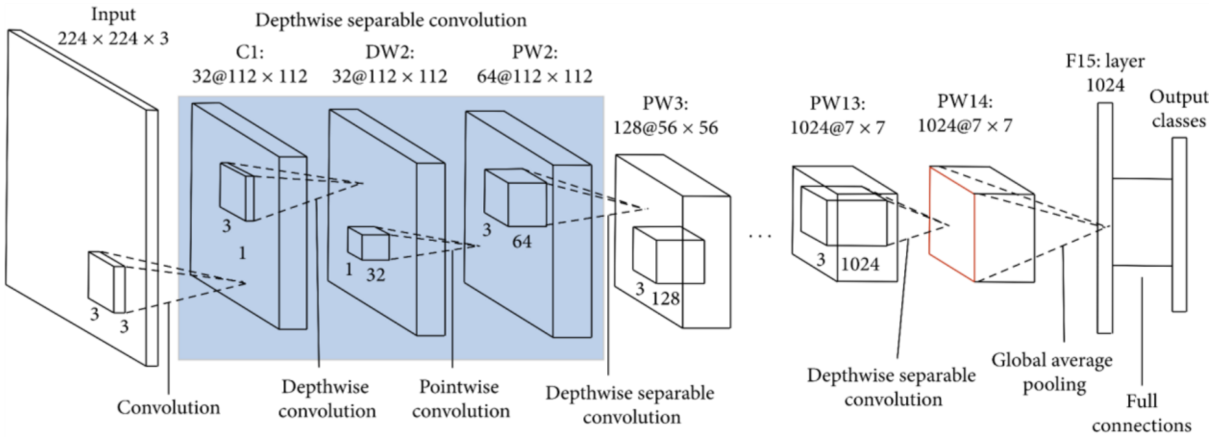


Fig 3.10: Architecture of MobileNet (Web Source: https://www.hindawi.com/journals/misy/2020/7602384/fig1/)

4. Results & Analysis:

The dataset was trained and evaluated on three different classification models. Results of the experiments conducted in this project are shared below in this section.

4.1 Training Details:

The dataset was divided into three splits, train, validation, and test in the ratio 0.7:0.1:0.2 for two classes, meteor, and plane. Fig shows data distribution across three splits. It can be observed that there is a class imbalance as images belonging to planes were almost three times more than the meteor images, so to handle this imbalance appropriate evaluation metrics were used.

The data was uploaded on Google drive to be accessed through Google Colaboratory as it enables usage of GPU during the training.

Training was done for 25 epochs for all three models, with batch size 8. Cross Entropy Loss was used as the loss function and Adam Optimizer was used with standard learning rate of 0.001

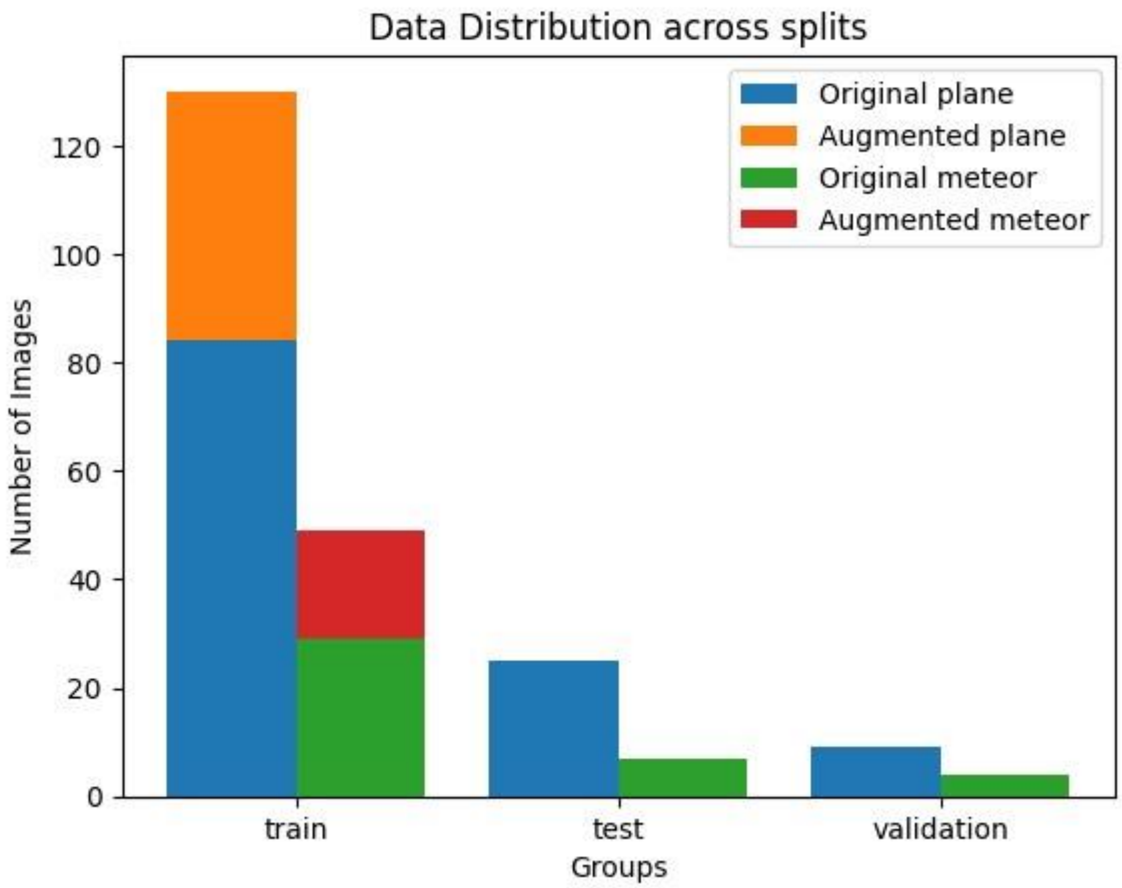


Fig 4.1: Data distribution (original and augmented images) across all three splits

Performance Evaluation:

It is integral to choose appropriate metrics to evaluate a model's performance on training and unseen data. Classification results were evaluated on ROC-AUC, Macro F1 score, Precision and Recall. Since our dataset was highly imbalanced, using Accuracy as the metric would have given biased results [23].

The results of classification were mapped as binary output as there were two classes, 0 representing plane category and 1 representing meteor label. Classification scores for each class can be categorised through four possibilities, true positive (TP), false positive, true negative, and false negative. When the model predicts a positive class correctly, it is a true positive. If it marks a positive class as negative, then it is a false negative. If a negative class is labelled correctly in the prediction, then it is a true negative case, if not then it is false positive. Figure below gives better insight through a confusion matrix.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Fig 4.2 Confusion metrics used to map output of binary classification (Web source: https://glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/)

AUC curve evaluates performance of a classifier at different thresholds. It is a probability curve which tells how a good classifier distinguishes between different classes. Higher the curve, better is the performance of the classifier. AUC-ROC curve plots False Positivity Rate against True Positivity Rate. Formulae below shows how FPR and TPR are calculated.

$$\text{TPR/Recall/Sensitivity} = \frac{TP}{TP+FN}$$

F1 score is a ratio calculated over Precision and Recall. Macro F1 is calculated as average of individual classes' precision and recall, where Micro F1 is the sum of individual classes' true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs) of the model which is then placed in the F1 formula. F1 score is a metric which is calculated per class therefore is beneficial when working with class imbalance. F1 score can be aggregated through macro f1 and micro f1. Micro F1 gives equal weightage to all the data points in both classes, whereas macro f1 gives importance to each class equally. Therefore, in our case, we worked with macro f1 as it can evaluate performance of each class objectively.

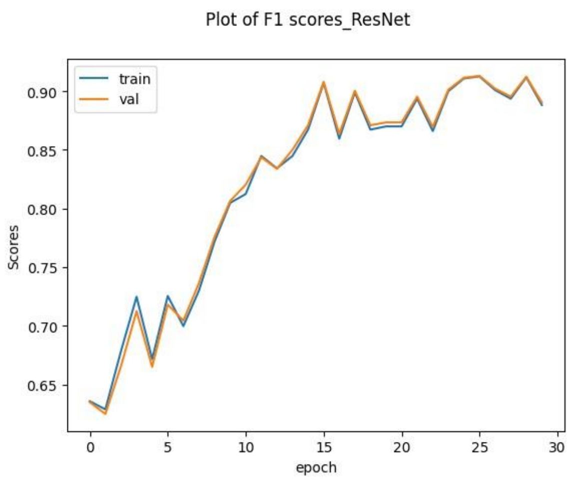
$$\text{Precision} = \frac{TP}{TP+TP}$$

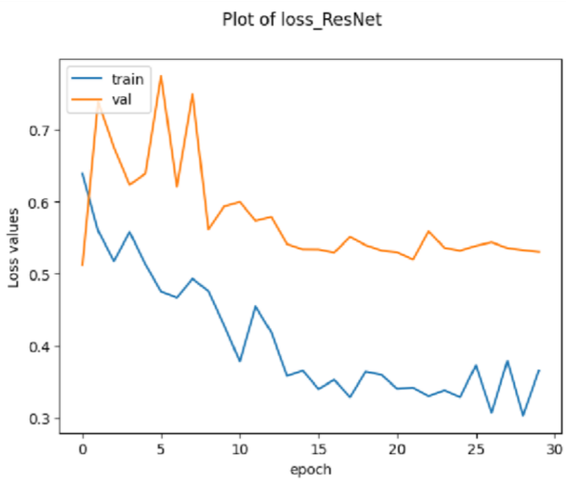
$$\text{F1 Score} = \frac{2(Precision*Recall)}{Precision+Recall}$$

Precision and Recall were also incorporated in validation results. Precision measures true positives of all the positive predictions. Higher the value of precision, better it is, ideal being 1, which suggests that a good classifier predicts fewer false positives. It is calculated by dividing the number of true positives by the sum of the true positives and the false positives.

Recall is used to calculate true positives from all the actual positives in the data. Like precision, recall also has 1 as max value. It is calculated by dividing the number of true positives by the sum of the true positives and the false negatives. Higher value of recall means the classifier is good at avoiding false negatives.

Results:





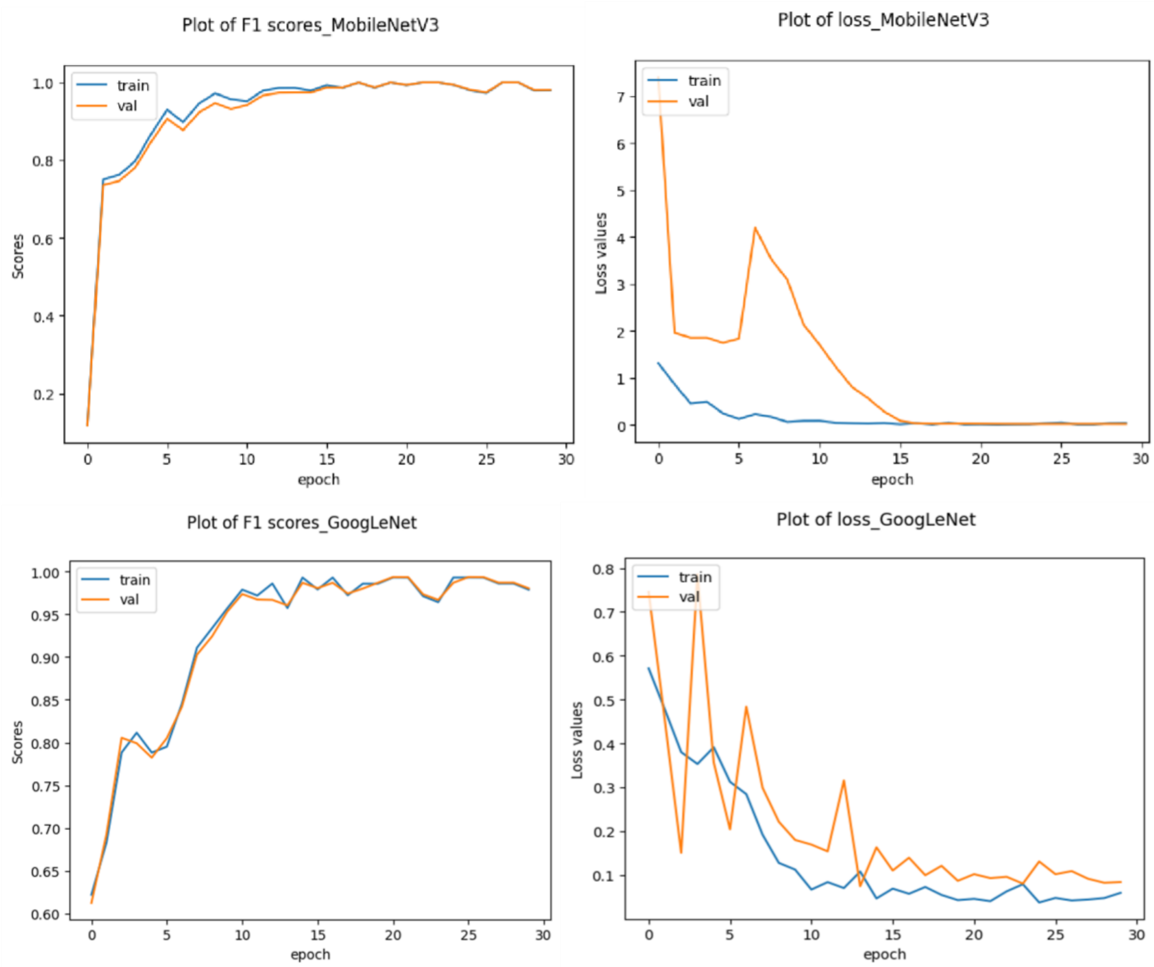


Fig 4.3: Training & Validation plots of all three networks

Models	**Macro F1**	**Precision**	**Recall**	**AUC ROC**
ResNet50	0.67	0.87	0.69	0.69
MobileNetV3	0.83	0.94	0.78	0.79
GoogLeNet	0.76	0.93	0.714	0.7

Table 4.1 Test results of models and their respective evaluation metrics

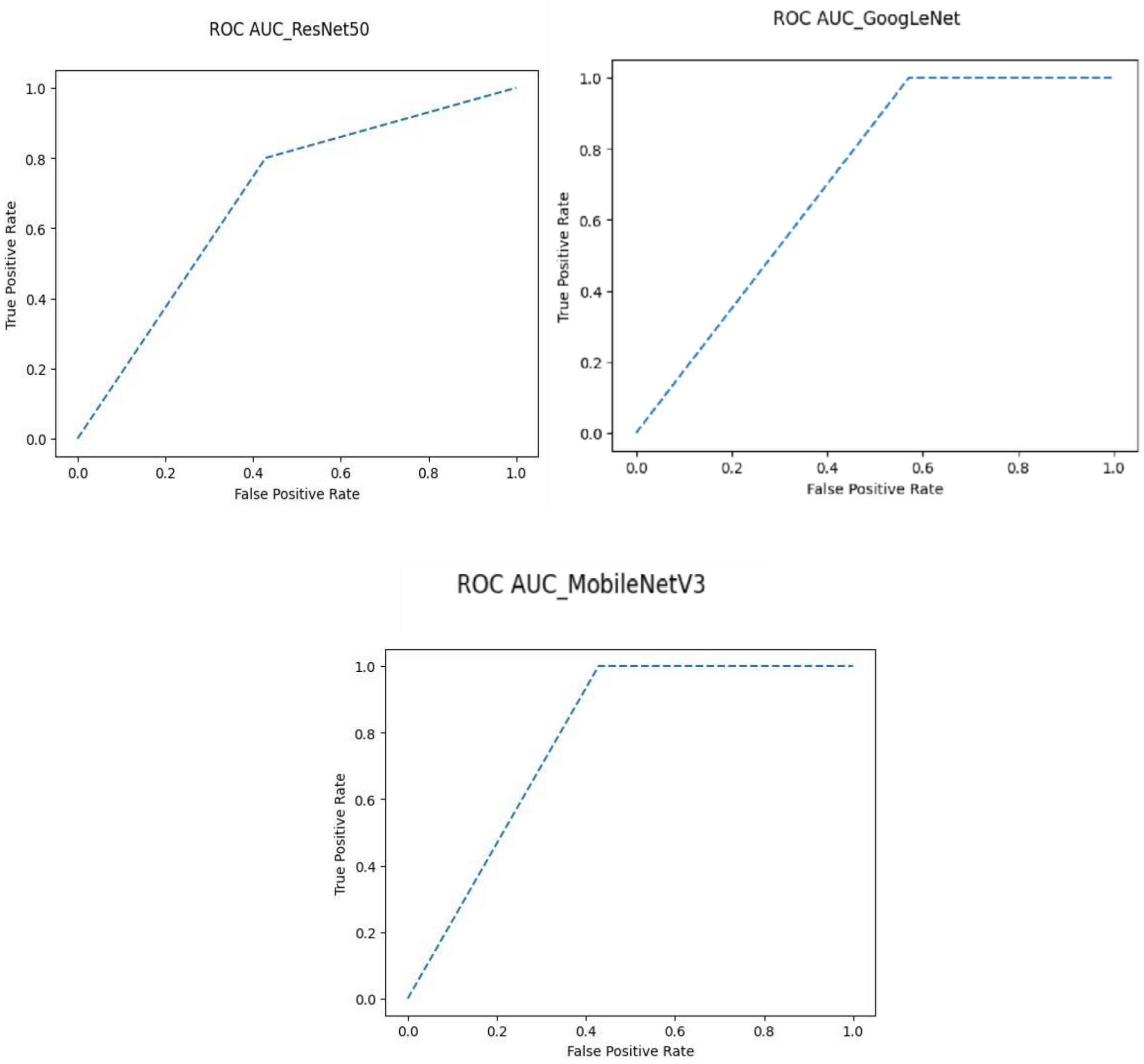


Fig 4.5 ROC-AUC plots on test set

	precision	recall	f1-score	support
1	0.87	0.80	0.83	25
0	0.44	0.57	0.50	7
accuracy			0.75	32
macro avg	0.66	0.69	0.67	32
ıeighted avg	0.78	0.75	0.76	32

Classification report of ResNet 50

	precision	recall	f1-score	support
0	1.00	0.57	0.73	7
1	0.89	1.00	0.94	25
accuracy			0.91	32
macro avg	0.95	0.79	0.84	32
weighted avg	0.92	0.91	0.90	32

Classification report of MobileNetV3

	precision	recall	f1-score	support
0	1.00	0.43	0.60	7
1	0.86	1.00	0.93	25
accuracy			0.88	32
macro avg	0.93	0.71	0.76	32
weighted avg	0.89	0.88	0.85	32

Classification report of GoogLeNet

Fig 4.6: Classification reports of the networks on test set

Discussion:

Deep learning models are data hungry and require large volumes of data to be trained to get good results. Our model was data deprived as we were only able to get 158 images from the observatory. In such cases, we need shallow models to perform feature extraction. The models we used had layers ranging from 22 in GoogLeNet to 50 in ResNet50 (Table 4.2). There needs to be an optimal number of layers to perform convolution, too less of layers do not have enough complexity and fine visual details are not captured, whereas too many layers tend to perform overfitting and lower the results. Not only the number of layers makes a difference, design and internal architecture plays an important role in converging the results. As discussed in the previous section, MobileNetV3 is specifically designed to work best on small applications and can therefore work well on small datasets as well [24]. Therefore, we saw better performance by MobileNetV3 in training as well as during final testing.

Model	No of Layers
ResNet50	50
GoogLeNet	22
MobileNetV3	28

Table 4.2 Number of layers in the experimented models

Generally, all three models show increasing trend for f1 score and decreasing trend for loss, but it can be observed that there is a lot of fluctuation between the epochs. One of the reasons for high fluctuations in validation results and low values represents overfitting over small amounts of data [25]. MobileNetV3 (Table 4.1) adequately performed well on the given dataset suggesting its number of layers and internal blocks are appropriate for small imbalanced data classification. Not only it gave the highest value of AUC ROC results among all, the trend of plot is also in the upwards trend and with minimum fluctuation like a good curve which should be ideally achieved.

Table 4.1 shows models performance on the final test set. There were 32 images in the test set with arbitrary selection of labels by the model, 1 representing plane class and 0 representing meteor. All the three models showed positive promising results, with MobileNetV3 proving to be good at classification for all the four metrics, and ResNet50 scoring lowest value of all. It was noted that all the models had generally average recall scores around 70% suggesting that models' performance was not that exemplary when identifying positive instances. Precision values for all the models were above 80% which meant that correctness of predictions was at above-average standing.

AUC ROC and classification reports for the models are shown in fig 4.5 and fig 4.6 respectively. Higher curve means higher value for area under the curve for the plot. The closer the AUC ROC curve is to the top-left corner of the graph, the better the model is at distinguishing between the two classes, through this we can observe MobileNetV3 was best among all at distinguishing between both classes, followed by GoogLeNet and ResNet50. Classifications reports tell us about class wise performance of the algorithms. For all the three models, 0 label i.e planes had higher scores than meteors with highest score for plane at 1 for recall and precision in GoogLetNet and MobileNetV3. This difference is quite intuitive considering the availability of data points for each class. Images labelled as plans had more representation therefore their feature extraction and learning were better than the meteor images.

5. Conclusion:

Classification of meteors against plane in sky images was targeted in this project. This was achieved through experimentation and analysis of three different classification models, ResNet50, GoogLeNet and MobileNetV3. This project proved to be a multifaceted problem, as it was not only about classification, but also about handling small imbalanced datasets, and the choice of classification network when dealing with such constraints. Models were evaluated on four different metrics, F1 macro, Precision, Recall & AUC-ROC score and not on accuracy to avoid bias of imbalance in the calculation. MobileNetV3 performed best at classification followed by GoogLeNet and then ResNet50. The results also shed light on the importance of number of layers, model depth, and ithe internal architectural details used in each model and how it was related with the performance. There has not been much work done in application of computer vision in astrophysics. The results came out to be positively distinctive and prove promising in exploring this domain. In future work, algorithms and our current pipeline can be honed to yield better results by incorporating more data, as an inadequate amount of data was a limiting factor in upgrading the results.

References:

1. Vítek, S., & Nasyrova, M. (2017). Real-Time detection of sporadic meteors in the intensified TV imaging systems. *Sensors*, *18*(1), 77. https://doi.org/10.3390/s18010077

2. Indolia, S., Goswami, A. K., Mishra, S., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, *132*, 679–688. https://doi.org/10.1016/j.procs.2018.05.069

3. Cecil, D., & Campbell-Brown, M. (2020). The application of convolutional neural networks to the automation of a meteor detection pipeline. *Planetary and Space Science*, *186*, 104920. https://doi.org/10.1016/j.pss.2020.104920

4. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for LargeScale Image Recognition. arXiv preprint arXiv:1409.1556.

5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770-778).

6. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 2818-2826).

7. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML) (pp. 6105-6114).

8. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems (NIPS) (pp. 91-99).

9. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 779-788).

10. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 3431-3440).

11. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:1408.5882.

12. Vítek, S., & Nasyrova, M. (2017b). Real-Time detection of sporadic meteors in the intensified TV imaging systems. *Sensors*, *18*(1), 77. https://doi.org/10.3390/s18010077

13. *Computer Vision for astronomical image analysis*. (2021, December 17). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9687023

14. Solares, J. R. A., Raimondi, F., Zhu, Y., Rahimian, F., Canoy, D., Tran, J., Pinho-Gomes, A., Payberah, A. H., Zottoli, M., Nazarzadeh, M., Conrad, N., Rahimi, K., & Salimi-Khorshidi, G. (2020). Deep learning for electronic health records: A comparative review of multiple deep neural architectures. *Journal of Biomedical Informatics*, *101*, 103337. https://doi.org/10.1016/j.jbi.2019.103337

15. *Data augmentation for improving deep learning in image classification problem*. (2018, May 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8388338

16. Xu, Y. (2020, October 3). *WeMIX: How to Better Utilize Data Augmentation*. arXiv.org. https://arxiv.org/abs/2010.01267

17. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition,"

18. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90

19. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

20. Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

21. A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks by He, Zhang, and Sun (2017).

22. *Searching for MobileNetV3*. (2019, October 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9008835

23. MobileNetV3: Inverted Residuals and Linear Bottlenecks" by Howard et al. (2019). https://arxiv.org/abs/1905.02244.

24. Gütter, J., Kruspe, A., Zhu, X. X., & Niebling, J. (2022). Impact of Training Set Size on the Ability of Deep Neural Networks to Deal with Omission Noise. *Frontiers in Remote Sensing*, *3*. https://doi.org/10.3389/frsen.2022.932431