BASKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF ELECTRICAL ELECTRONICS ENGINEERING
MASTER OF SCIENCE IN ELECTRICAL ELECTRONICS ENGINEERING

VISION TRANSFORMER NETWORK IMPLEMENTATION FOR MULTI-LABEL IMAGE CLASSIFICATION

BY

EMRE AKKAŞ

MASTER OF SCIENCE THESIS

ANKARA – 2024

BASKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING
DEPARTMENT OF ELECTRICAL ELECTRONICS ENGINEERING
MASTER OF SCIENCE IN ELECTRICAL ELECTRONICS ENGINEERING

VISION TRANSFORMER NETWORK IMPLEMENTATION FOR MULTI-LABEL IMAGE CLASSIFICATION

BY

EMRE AKKAŞ

MASTER OF SCIENCE THESIS

ADVISOR

DOÇ. DR. SELDA GÜNEY

ANKARA – 2024

BASKENT UNIVERSITY
INSTITUTE OF SCIENCE AND ENGINEERING

This study, which was prepared by Emre AKKAŞ, for the program of Electrical Electronics Engineering, has been approved in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in Electrical Electronics Engineering Department by the following committee.

Date of Thesis Defence: 22 / 08 / 2024

Thesis Title: Vision Transformer Network Implementation for Multi-label Image Classification

Examining Committee Members	**Signature**
Prof. Dr. Hamit Erdem, Başkent Üniversitesi	.....
Dr. Öğr. Üyesi Koray Açıcı, Ankara Üniversitesi	.....
Doç. Dr. Selda Güney, Başkent Üniversitesi	.....

APPROVAL

Prof. Dr. Dilek ÇÖKELİLER SERDAROĞLU

Director of the Institute of Science and Engineering

Tarih: 22 / 08 / 2024

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORİJİNALLİK RAPORU

Tarih: 22 / 08 / 2024

Öğrencinin Adı, Soyadı : Emre Akkaş

Öğrencinin Numarası : 22110441

Anabilim Dalı : Elektrik-Elektronik Mühendisliği

Programı : Tezli Yüksek Lisans

Danışmanın Unvanı/Adı, Soyadı : Doç. Dr. Selda Güney

Tez Başlığı : Çoklu Etiketli Görüntü Sınıflandırması İçin Görüntü Dönüştürücü Ağ Uygulaması

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 49 sayfalık kısmına ilişkin, 29 / 08 / 2024 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %14'tür.

Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

"Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını" inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası :

Onay

Öğrenci Danışmanı

Doç. Dr. Selda Güney

Tarih: 22 / 08 / 2024

I dedicate this thesis to each of our esteemed engineers, collaborators, and institutions in our country who are conducting and advancing work in the field of space exploration.

Emre AKKAŞ

ANKARA - 2024

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Doç. Dr. Selda GÜNEY for her contributions to finish this thesis successfully. Her support and positivity assisted me to write this thesis.

I would also like to thank my mother Sevgi AKKAŞ for always believing in me since my childhood, providing support in every matter and trusting me unconditionally; my father Yücel AKKAŞ for always having complete faith in me and being a source of inspiration to me with his intelligence in the fields of art and engineering; my sister Sevde AKKAŞ ASLAN for working tirelessly even under the toughest conditions and showing that the strength to overcome everything is actually within us; and my wife Pelin AKKAŞ for always passionately committing to the things she feels deep in heart and instilling that passion in me as well.

ABSTRACT

Emre AKKAŞ

VISION TRANSFORMER NETWORK IMPLEMENTATION FOR MULTI-LABEL IMAGE CLASSIFICATION

Başkent University Institute of Science and Engineering

Department of Electrical and Electronics Engineering

2024

Swift process in technology and widespread availability of low-cost internet have led to a substantial rise in data volume in remote sensing, especially for high-resolution and very-high resolution images. Still, these images contain more complex information, and it is not appropriate to analyze the images using a solitary scene-level label while ignoring the distinct features provided by other labels in the images. In multi-label image classification applications, multiple labels are assigned to an image, reflecting various objects or features present in the scene. The classification of these images is critically important for monitoring environmental changes over large geographical areas, disaster management, urban planning, agriculture and forestry management, natural resource conservation, and military intelligence. Nowadays, many methods are used in such image classification problems, primarily deep learning algorithms. In this thesis, advanced neural networks are explored and evaluated for Multi-label AID dataset which contains 3000 images and 17 different labels; AlexNet, VGG16, DenseNet-201, Inception-v3 and ConvNeXt as the CNN models, ViT, SwinT as transformer models and MaxViT as the hybrid model that initially contains both CNN and transformer network. A fusion network that combines the strengths of a CNN and transformer model is created and applied to harness the spatial feature extraction capabilities of the CNN model and the global context understanding of the transformer model. Two distinct schedulers, OneCycleLR and ReduceLROnPlaetau, and two different loss functions, ASL and BCEWithLogitsLoss, are employed for each model to systematically evaluate their impact on model performance. The window-based MaxViT algorithm, which has not been previously applied to the Multi-label AID dataset in the current literature, has been evaluated. This algorithm has demonstrated superior performance on this dataset, significantly outperforming existing models and setting a new benchmark with an mAP of 84.98%.

KEYWORDS: Multi-label AID, Multi-label Image Classification, MaxViT, Asymmetric Loss, OneCycleLR

ÖZET

Emre AKKAŞ

ÇOKLU ETİKETLİ GÖRÜNTÜ SINIFLANDIRMASI İÇİN GÖRÜNTÜ DÖNÜŞTÜRÜCÜ AĞ UYGULAMASI

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

2024

Teknolojideki hızlı gelişmeler ve düşük maliyetli internetin yaygın olarak bulunabilirliği, uzaktan algılama alanında, özellikle yüksek ve çok yüksek çözünürlüklü görüntüler için veri hacminde önemli bir artışa yol açmıştır. Ancak, bu görüntüler daha karmaşık bilgiler içerdiğinden, görüntüleri yalnızca tek bir sahne seviyesindeki etiketle analiz etmek, diğer etiketlerin sağladığı belirgin özellikleri göz ardı etmek uygun değildir. Çok etiketli görüntü sınıflandırma uygulamalarında, sahnede bulunan çeşitli nesneleri veya özellikleri yansıtan birden fazla etiket bir görüntüye atanır. Bu görüntülerin sınıflandırılması, geniş coğrafi alanlarda çevresel değişiklikleri izlemek, afet yönetimi, şehir planlaması, tarım ve ormanlık alanların yönetimi, doğal kaynakların korunması ve askeri istihbarat gibi çeşitli alanlarda kritik önem arz etmektedir. Günümüzde bu tür görüntü sınıflandırma problemlerinde derin öğrenme algoritmaları başta olmak üzere birçok yöntem kullanılmaktadır. Bu tez çalışmasında, AlexNet, VGG16, DenseNet-201, Inception-v3 ve ConvNeXt gibi derin öğrenme yöntemlerinden olan Evrişimsel Sinir Ağları (Convolutional Neural Network, CNN) modelleri, ViT, SwinT gibi görsel dönüştürücü modelleri ve hem CNN hem de görsel dönüştürücü ağı içeren MaxViT hibrit modeli, 3000 adet resim ve 17 adet farklı etiket içeren Multi-label AID veri seti için incelenmiş ve değerlendirilmiştir. Bir CNN ve görsel dönüştürücü modelinin güçlü yönlerini birleştiren bir füzyon ağı, CNN modelinin mekansal özellik çıkarma yeteneklerini ve dönüştürücü modelinin küresel bağlam anlama yeteneklerini kullanmak amacıyla oluşturulmuş ve uygulanmıştır. Her model için iki farklı öğrenme oranı düzenleyicisi, OneCycleLR ve ReduceLROnPlateau, ve iki farklı kayıp fonksiyonu, ASL ve BCEWithLogitsLoss, sistematik olarak değerlendirilerek performansa etkileri incelenmiştir. Mevcut literatürde daha önce Multi-label AID veri setine uygulanmamış olan pencere tabanlı MaxViT algoritmasının değerlendirilmesi yapılmıştır.

Bu algoritma, bu veri setinde üstün performans sergileyerek mevcut modelleri önemli ölçüde geride bırakmış ve %84.98 mAP elde ederek yeni bir standart belirlemiştir.

ANAHTAR KELİMELER: Multi-label AID, Çok Etiketli Görüntü Sınıflandırma, MaxViT, Asymmetric Loss, OneCycleLR

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS AND ABBREVIATIONS

AID	Aerial Image Dataset
AP	Average Precision
ASL	Asymmetric Loss
BCE	Binary Cross Entropy
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DeepViT	Deep Vision Transformer
DeiT	Data-efficient Image Transformer
DINO	Self **Di**stillation With **No** Labels
DL	Deep Learning
DSViT	Dynamically Scalable Vision Transformer
EsViT	Efficient Self-supervised Vision Transformer
FFN	Feedforward Network
FFNN	Feed-Forward Neural Network
FLOP	Floating-point operations
FN	False Negative
FP	False Positive
GB	GigaByte
GELU	Gaussian Error Linear Unit
GIS	Geographic Information system
GNN	Graphical Neural Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LDBST	Lightweight Dual-branch Swin Transformer
LD-GCN	Label-Driven Graph Convolutional Network
LR	Learning Rate
LRGCN	Label Regocnition Graph Convolutional Network
LSTM	Long-Short Term Memory
mAP	Mean Average Precision
MAT	Masked-Attention Transformer

MaxViT	Multi-Axis Vision Transformer
MHSA	Multi-Head Self Attention
MLIC	Multi-label Image Classification
MLP	Multi-Layer Perceptron
MSE-Net	Master-slave Encoding Network
OA	Overall Accuracy
OneCycleLR	One Cycle Learning Rate
PVT	Pyramid Vision Transformer
ReLU	Rectified Linear Unit
RESISC	Remote Sensing Image Scene Classification
ResNet	Residual Network
RGB	Red Green Blue
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
RS	Remote Sensing
RSSCN	Remote Sensing Scene Classification Network
SCViT	Spatial-channel Vision Transformer
SE	Squeeze-and-excitation
SLIC	Single Label Image Classification
SNN-VGG	Spiking Neural Network – Visual Geometry Group
SRBM	Semantic Relation-Building Module
SSM	Semantic Sensitive Module
SOTA	State-of-the-art
SwinT	Swin Transformer
T2T	Tokens-to-Token
TN	True Negative
TP	True Positive
UCM	UC Merced
VGG	Visual Geometry Group
ViT	Vision Transformer

1. INTRODUCTION

Space has always been a wonder of humanity. Valuable explorations were started by Nicolaus Copernicus in the 16th Century. He introduced the heliocentric model which suggested that Earth and other planets revolved around the sun. In the following century, Galileo Galilei invented the telescope and observed many planets [1]. Mankind's curiosity about space has continued with looking at the sky and exploring space and continues to increase today. With the developing technology during the 1930s, the view of the world from space has also become important and imperative in terms of filling the gaps in scientific studies and military fields.

Swiss physicist and inventor Auguste Piccard, famous for his contributions to high-altitude ballooning, reached an altitude of approximately 15.7 kilometers in a pressurized gondola suspended from a balloon. During his ascent in 1931, he reported seeing the curvature of the Earth and the dark sky above. Even if he didn't take photographs, his observations were significant for early space exploration [2]. Same as Piccard, in 1935, Albert W. Stevens and Orvil A. Anderson captured photographs of the Earth's curvature and the blackness of space above by the high-altitude balloon launched by the U.S Army [3].

After World War 2, the United States has launched a rocket called as V-2 which was taken from Germany and was used for scientific and military purposes as well as space exploration. It was launched on October 24 1946, and reached an altitude of 112 kilometers. Considering the space border to be as 100 km, the first image from the space was captured by 35 mm the black-and-white motion picture camera [4].

The success of the V-2 rocket and the subsequent development of rocket technology led to the launch of various satellites and scientific instruments into space. Despite marking the beginning of the space age and initiating a series of satellite missions by the Soviet Union in 1957, Sputnik 1 did not carry any cameras or imaging equipment for capturing photographs. Discoverer 14, launched by the United States, became the first space-based Earth observation system, paving the way for advanced reconnaissance satellites [5].

Thousands of satellites have been launched and put into orbit since then. Numerous satellites have completed their life cycles and are no longer operational. Some remain in orbit as space debris, while others have been deorbited and burned up in the Earth's

atmosphere. Most of them have served for different purposes and equipped with various sensors, enabling remote sensing of the planet’s surface features. The term "remote sensing" was initially originated by the United States Naval Research Officer, Ms. Evelyn Pruitt, during the 1950s [6]. In contemporary usage, it commonly refers to the scientific and artistic practice of identifying, observing, and quantifying an object without direct physical interaction. It is highly significant across multiple domains such as agriculture [7], urban planning [8], forestry [9], geospatial analysis [10] etc. to gather valuable information about the Earth's features, conditions, and changes over time.

In agriculture, remote sensing facilitates precision farming through the monitoring of crop health, the optimization of irrigation, and early pest and disease detection. Urban planning benefits from remote sensing's land utilization and surface characteristics analysis, contributing to informed zoning decisions and infrastructure planning. In forestry, it plays a vital role in forest management, from assessing forest health to detecting deforestation. Environmental monitoring relies on remote sensing for ecosystem health assessments and rapid response to natural disasters. Geospatial analysis utilizes remote sensing data for creating spatial databases, maps, and supporting GIS applications. Across these areas, remote sensing empowers informed decision-making, resource management, and scientific research, contributing to sustainability, efficiency, and public safety.

The rapid process in technologies and cheap internet connection has increased data too much. Huge amount of imagery shall be analyzed to extract meaningful information from the domains mentioned above, the well-known method for this is called as image classification. It is a process of categorizing pixels in an image into predefined classes, for instance different types of land cover. A single label is assigned to each image with the traditional classification techniques. But this can be limiting factor where multiple features exist in a complex environment. On the other hand, assignment of multiple labels to a pixel or single image is possible with multiple image classification, enabling more accurate and detailed analysis of satellite imagery. This advanced method is particularly useful in remote sensing applications.

1.1. Aim of the Study

Traditional image classification assigns one tag to identify an image, while in multi-label image classification (MLIC), several tags are used to describe different aspects or components of the scene.

This study mainly aims to explore and assess sophisticated neural networks for multi-label image classification, focusing on the Multi-label AID dataset, which is not very well-known and has shown relatively low classification accuracy in earlier studies. This thesis involves the implementation and optimization of CNN-based, vision transformer, fusion and hybrid networks. Through comprehensive experimentation and assessment, this thesis aims to identify the most effective model architecture and configuration for the Multi-label AID dataset and highlighting the superiour performance of the MaxViT algorithm. This thesis seeks advancement in deep learning and multi-label image classification by providing insights into the application and effectiveness of the SOTA architectures and optimization techniques.

1.2. Thesis Outline

This thesis is categorized into 6 different sections as follows:

In the first section, the endeavors of humanity in the field of space, from past to the present day, is summarized. The works carried out in line with the aim and the organization of the thesis is presented.

In the second section, a concise overview of image classification is provided.

In the third section, other works in the literature related to image classification and deep learning applications are discussed.

In the fourth section, content of the dataset that is given. In addition, the theory behind the various networks, schedulers, loss functions and activation functions are elaborated. The mathematical equations of the evaluation metrics are given.

In the fifth section, the various experiments that are conducted for this study and the obtained results are presented and evaluated.

In the final section, the inadequacies of the dataset, the relative low success rates, and how more succesful results are achieved with previously unexplored model are explained.

2. IMAGE CLASSIFICATION

Human beings can effortlessly carry out image classification by examining the components visible in the image. This method involves tagging or categorizing a specific element. For example, a key on a table might be identified merely as 'key', while ignoring the table itself. In the field of computer vision, this is recognized as the problem of assigning either a single or multiple categories to an entire picture.

2.1. Single Label Image Classification

Classification methods that assign only one category to an image deal with pictures that are linked to a single category from a limited collection of non-overlapping categories, also known as single label image classification (SLIC). It has attracted extensive attention in different applications such as object detection [11], image retrieval [12] etc. Many existing image classification studies [13], [14], [15], [16], [17] focused on single label remote sensing classification. However, single label on the real-world image is often insufficient to completely describe the content. It considers scenarios where every image has a single label. The cases that an image is connected to multiple labels is disregarded.

2.2. Multi-label Image Classification

Given that high-resolution images display various types of categories within a single remote sensing image, a single label is inadequate to describe the content or describe the rich information of an image which can contain semantically complex substance [18]. Instead of just one label, image annotations have grown more intricate, with visual data now being tagged with more than one. Multi-label image classification is more realistic approach when it comes to associating various meaningful concepts. However, when compared to SLIC, the classification of multi-label turns into more challenging due to its additional complexity. Number of relevant features and samples bring the complexity for the SLIC, where MLIC task has an extra complexity along the target aspect as discussed by Bogatinovski et al. [19]. But the challenges are not limited with the model complexity for MLIC, there are also more challenges such as class imbalance [20], correlated labels [21], computational complexity [22], data annotation [23], overfitting [24], thresholding [25], scalability [26].

3. RELATED WORKS

In this chapter, some studies and benchmark algorithms are reviewed for many deep learning and vision transformer applications. Many of the remote sensing applications have benefited from the progress, such as traffic monitoring [27], surface morphology analysis [28], urban management [29]. For the purpose of evaluating and analyzing the remote sensing images, different datasets are needed. There are plenty of multi-class datasets available online such as EuroSAT [30], RSSCN7 [31], UC Merced (UCM) [32], AID [33], NWPU-RESICS45 [34] and so on.

CNNs have been dominant approach for image understanding tasks, based on their superior performance on classification problems and this success has extended to many other image understanding tasks. ImageNet [35] dataset played a vital role in their success due to availability of a large training set. The evolution of the cutting-edge on the ImageNet dataset demonstrates the advancements with CNN architectures and learning [36], [37]. A rising focus has emerged on architectures employing attention mechanisms with convolution networks [38]. Several attempts have been made to use transformers on image classification, but the performance was not as successful as convnets. Nonetheless, hybrid architectures integrate transformers and convnets and exhibited notable results [39]. Vision Transformers (ViT) [40] have achieved SOTA results on ImageNet without the use of convolution.

Rangel et al. [41] performed a land cover image classification on EuroSAT dataset, which has 10 classes and geo-referenced images, and used many influential deep learning architectures. The evaluation metrics of this study is accuracy, and they achieved 99% accuracy on the dataset with MaxViT [42] algorithm.

Li-Ye et al. [43] proposed a method called as SNN-VGG-15 (TF-reset) and reached to 94.21% overall accuracy (OA) in RSSCN7 dataset. MSE-Net [44] surpassed this metrics and achieved to 97.76% by extracting higher-level semantic features.

After the Vision Transformer (ViT) [40] and Swin Transformer (SwinT) [45] are published and many studies have been performed with these vision transformers, Kaselimi et al. [46] implemented Vision Transformer to take advantage of the self-attention mechanism.

DSViT, introduced by Wang et al. [47], was designed to address the challenges related to the limitations of extracting comprehensive information using individual convolutional models and the constraints posed by high computational costs. This is achieved by developing an attention mechanism that adjusts dynamically and combines convolutional and transformer features. The model reached a mean average precision (mAP) of 93.44% and an accuracy of 96.19% on the UC Merced dataset.

SCViT is developed by Lv et al. [48] that is considering the contribution of distinct channels and considers geometric information in the classification token. This method generates tokens, introduces lightweight channel attention, models global interactions and uses a multilayer perceptron. The model's OA is 1.27% less than that of the Master-slave Encoding Network on the AID dataset [44].

LDBST [49] was published which aims an RS scene classification by employing a dual-branch structure. One part is fed into a ViT branch where the other part uses a simple-structured CNN branch. In UC-Merced, AID and NWPU-RESICS45 datasets, the proposed method achieved 99.52%, 96.84% and 94.36 accuracy respectively. But distinguishing high-frequency details from low-frequency details is not being considered which impacts the accuracy of interpretation to a certain extent.

DCNNs ([16], [17], [50], [51]) extract high level semantic features. However, these networks are mostly used for the single label remote sensing applications. Multiple semantic labels are not being considered, nor are the dependencies between labels. Hua et al. [52] utilized attention-based network to extract detailed semantic feature maps and used LSTM network to generate structured multiple object labels.

Wang et al. [53] introduced a CNN-RNN framework for multi-label image classification tasks. In this approach, the RNN follows the CNN to obtain a joint representation of the image and its labels, which is then used to produce label predictions.

Yeh et al. [54] created a deep latent space model designed to handle label domain data. They also introduced a loss function that responds to label correlations and connects features effectively.

ViT [55] can be used to capture contextual features, but it involves high computational demands and has limited capacity for learning. The images are segmented into patches of uniform size, and each patch is considered a separate token. The approach primarily

emphasizes capturing global information within the spatial domain. When compared to CNN algorithms, ViT is also highly dependent on a lot of reference data for the training.

Hassani et al. [56] presented the Neighborhood Attention Transformer, which reduces computational requirements by utilizing a flexible attention window.

PVT was introduced by Wang et al. [57] to address the shortcomings of ViT in dense prediction tasks. This backbone network is designed without convolution layers. It demonstrates efficiency in various dense prediction tasks, such as semantic segmentation and object detection. The method introduces a progressively shrinking pyramid structure along with a spatial-reduction attention layer to achieve multi-scale feature maps and high-resolution outputs while maintaining limited computational overhead. This allows the model to effectively manage dense prediction tasks by producing multi-scale feature representations. Furthermore, it represents the first hierarchical architecture for Vision Transformers (ViT).

Since the computational demands of the original PVT is relatively large while processing high-resolution images, PVTv2 [58] was introduced to improve the first version by introducing depth-wise convolution, efficient attention and overlapping patch embedding.

Swin Transformer (SwinT) was published by Lie et al. [45] which can act as an adaptable framework for various tasks such as dense prediction and image classification. Non-overlapping patches (tokens) of the given image are split by using a patch partitioning module. Linear embedding is applied to map the patch sequence into an arbitrary dimension. Several Swin Transformers blocks follow to apply self-attention. It should be noted that the feature map resolutions are similar to the ones in typical convolutional architectures like ResNet [50], which means that SwinT can replace ResNet backbone networks efficiently in computer vision tasks.

Self-supervised approaches are also investigated for Vision Transformers. DINO [59] demonstrated a unique phenomenon that although ViTs are not trained with pixel-level guidance, they can automatically identify the non-essential pixels. This phenomenon is not observed neither in fully supervised ViTs nor CNNs. EsViT was proposed by Li Et al. [60]. It introduces a multi-stage design and uses DINO for self-supervision. In addition to DINO, additional extra patch-level self-supervision techniques are incorporated.

When compared to different attention-based transformers architectures, MaxViT has been observed as the best performer by Tu et al. [42]. It has outperformed models like Cross-ViT [61], DeepViT [62], DeiT [63], T2T [64] etc. with a top-1 accuracy of %85.2. It integrates the strengths of improved CNNs and attention mechanisms into a new 'base block.' The base block consists of multiple components, including the 'MBConv' block that utilizes the SE module. This is followed by a multi-axis attention block, designed specifically to capture both local and global pixel relationships.

MaxViT has advantage of learning both local and global features simultaneously, when compared to ViT. As a central element of ViT, self-attention gathers long-range dependencies within input tokens. However, self-attention has computational complexity which is quadratic complexity with token lengths. Window self-attention [45] executes attention on non-overlapped windows, demands the cyclic shift operation with the aim of communicating information across windows. Since MaxViT is a Window-based Vision Transformer, it has improved the efficiency of ViT resulting in linear complexity of self-attention proportional to token length. Spatial reduction attention can represent global interactions between tokens from coarse features, whereas window self-attention identifies local features within windows. Both global and local information are complementary. Incorporating these features into model blocks leads to modeling ability and better feature representation [65].

The originators of Multi-label AID dataset, Hua et al. [66], created an attention-aware label relational reasoning network and achieved 88.72% CF1 score (per-category F1-Score) on their own dataset. Li et al. [18] utilized both visual and spatial information, combining CNN and GNN, and obtained 88.64% CF1 score. Tan et al. [67] proposed a network that contains two models: SSM and SRBM. SSM captures features using transformer for the purpose of extracting semantic attentional regions from visual attributes by DCNN. SRBM uses the output of SSM to generate the relation matrix and this network achieved 89.97% CF1 score. Wu et al. [68] achieved 92.81% CF1 score by presenting a Transformer-based framework; SDM, Semantic Disentanglement Module, and MAT, Masked-Attention Transformer. Ma et al. [69] proposed LD-GCN. Intrinsic label associations are learned by the label-correlation matrix and fed into LRGCN which harnesses relationship between images and labels. This network not only achieved 92.81% CF1 score, but also 83.49% of mAP is obtained.

4. MATERIALS AND METHODS

4.1. Dataset

The original single-label AID dataset was developed by Xia et al. [33]. Google Earth imagery was the source of the high-resolution aerial images in the dataset with sizes of 3 x 600 x 600. A combined total of 10,000 images are grouped under 30 classes with spatial resolution ranging between 0.5 – 8m. The initial dataset was relabeled by Hua et al. [66] in 2020 and it has become a multi-label version of the original dataset with the same resolution of the images. This new dataset, multi-label AID, consists of 3000 images in total with 17 labels. Every image receives manual annotation, assigning up to 11 labels. Table 1 lists quantity of images in Multilabel-AID dataset for each predefined label. Figure 1 illustrates several examples from the dataset with their associated labels.

Table 1. Number of images for Multi-label AID dataset

Label	Quantity
Airplane	99
Bare soil	1475
Buildings	2161
Cars	2026
Chaparral	112
Court	344
Dock	271
Field	214
Grass	2295
Mobile home	2
Pavement	2328
Sand	259
Sea	221

Ship	284
Tanks	108
Trees	2406
Water	852

Figure 1. Sample images from the Multi-label AID dataset

Table 2 shows the label distribution for training, validation and test sets.

Table 2. Label Distribution Across Train, Validation and Test Sets

Label	Train	Validation	Test
airplane	62	17	20
bare-soil	934	237	304
buildings	1372	372	417
cars	1288	329	409
chaparral	56	19	37
court	212	57	75
dock	174	47	50

field	133	42	39
grass	1463	366	466
mobile-home	1	0	1
pavement	1488	382	458
sand	166	41	52
sea	143	34	44
ship	189	48	47
tanks	85	2	21
trees	1523	401	483
water	525	149	178

Quantity of the images are limited in Multi-label AID dataset. Data augmentation is adopted for the purpose of avoiding overfitting. Following image modifications are made to the original images respectively.

- Resizing to 256x256,
- Randomly cropping 224x224 patch,
- Randomly horizontal flipping,
- Randomly vertical flipping,

An example of the augmented image is given in Figure 2.

Figure 2. Representation of augmented data from an image of Multi-label AID dataset

4.2. Model Architectures

4.2.1. AlexNet

Alex et al. [36] presented a research paper for their model, AlexNet, and won ILSVRC image classification in 2012. AlexNet has 5 convolutional and 3 fully connected layers, totaling around 61 million parameters and over 600 million connections. ReLU activation functions are employed in every layer except the output layer, as illustrated in Table 1. During its initial introduction, AlexNet gained significant popularity in research due to its impressive track record of achieving low error rates. However, in recent years, its utilization has declined as more advanced neural networks have emerged. AlexNet, with its comparatively higher parameter count, lags behind these modern neural networks in terms of performance.

Table 3. AlexNet Architecture

Type	Filters	Filter Size/Stride	Padding	Output
Convolutional	96	11 x 11 / 4		55 x 55 x 96
Max Pooling		3 x 3 / 2		27 x 27 x 96
Convolutional	256	5 x 5 / 1	2	27 x 27 x 256
Max Pooling		3 x 3 / 2		13 x 13 x 256
Convolutional	384	3 x 3 / 1	1	13 x 13 x 384
Convolutional	384	3 x 3 / 1	1	13 x 13 x 384
Convolutional	256	3 x 3 / 1	1	13 x 13 x 256
Max Pooling		3 x 3 / 2		6 x 6 x 256
Fully Connected				4096
Fully Connected				4096
Fully Connected				1000
Softmax				

4.2.2. VGG16

VGGNet, devised in 2014 by Simonyan et al. [16] at Oxford University's Visual Geometry Group, aims to enhance classification accuracy by employing deeper neural networks. Various VGG network variants have been crafted for this purpose, differing in the number of layers. VGG networks come in 11, 13, 16, or 19-layer configurations, featuring varying convolutional layer counts. Each VGG network comprises three fully connected

layers. VGG-16, a specific convolutional neural network within the VGG family, boasts 13 convolutional layers, three fully connected layers, and a substantial parameter count of 138 million. It employs 224 x 224 RGB images as input. In the VGG-16 model, distinct blocks with varying numbers of convolutional layers, using 3 x 3 filters with a stride of 1, have been constructed. These blocks are interspersed with five max pooling layers using 2 x 2 filters and a stride of 2. Beyond the final convolutional layer, there are three fully connected layers, each yielding 4096, 4096, and 1000 outputs, culminating in a softmax layer as the ultimate layer [9]. Table 4 illustrates the architecture of VGG-16.

Table 4. VGG-16 Architecture

Type	Filters	Filter Size/Stride	Output
Convolutional	64	3 x 3 / 1	224 x 224 x 64
Convolutional	64	3 x 3 / 1	224 x 224 x 64
Max Pooling		2 x 2 / 2	112 x 112 x 64
Convolutional	128	3 x 3 / 1	112 x 112 x 128
Convolutional	128	3 x 3 / 1	112 x 112 x 128
Max Pooling		2 x 2 / 2	56 x 56 x 128
Convolutional	256	3 x 3 / 1	56 x 56 x 256
Convolutional	256	3 x 3 / 1	56 x 56 x 256
Convolutional	256	3 x 3 / 1	56 x 56 x 256
Max Pooling		2 x 2 / 2	28 x 28 x 256
Convolutional	512	3 x 3 / 1	28 x 28 x 512
Convolutional	512	3 x 3 / 1	28 x 28 x 512
Convolutional	512	3 x 3 / 1	28 x 28 x 512
Max Pooling		2 x 2 / 2	14 x 14 x512
Convolutional	512	3 x 3 / 1	14 x 14 x512
Convolutional	512	3 x 3 / 1	14 x 14 x512
Convolutional	512	3 x 3 / 1	14 x 14 x512
Max Pooling		2 x 2 / 2	7 x 7 x 512
Fully Connected			4096
Fully Connected			4096
Fully Connected			1000

Softmax			

4.2.3. DenseNet-201

DenseNet-201, as the name suggests, consists of 201 layers. In this structure, as seen in Figure 3, every layer is directly linked to all the other layers. In other terms, each layer takes input from every preceding layer and contributes output to every subsequent layer. One advantage of DenseNet-201 is that it encourages the reuse of features through these direct connections between layers. However, due to these connections between the layers, the network requires high computational resources, which is a disadvantage. Each layer in the network's design integrates outputs from all preceding layers within the same framework [10]. Such approach increases the variety of features. Due to these advantages, DenseNet-201 has been successful in classifying satellite images.

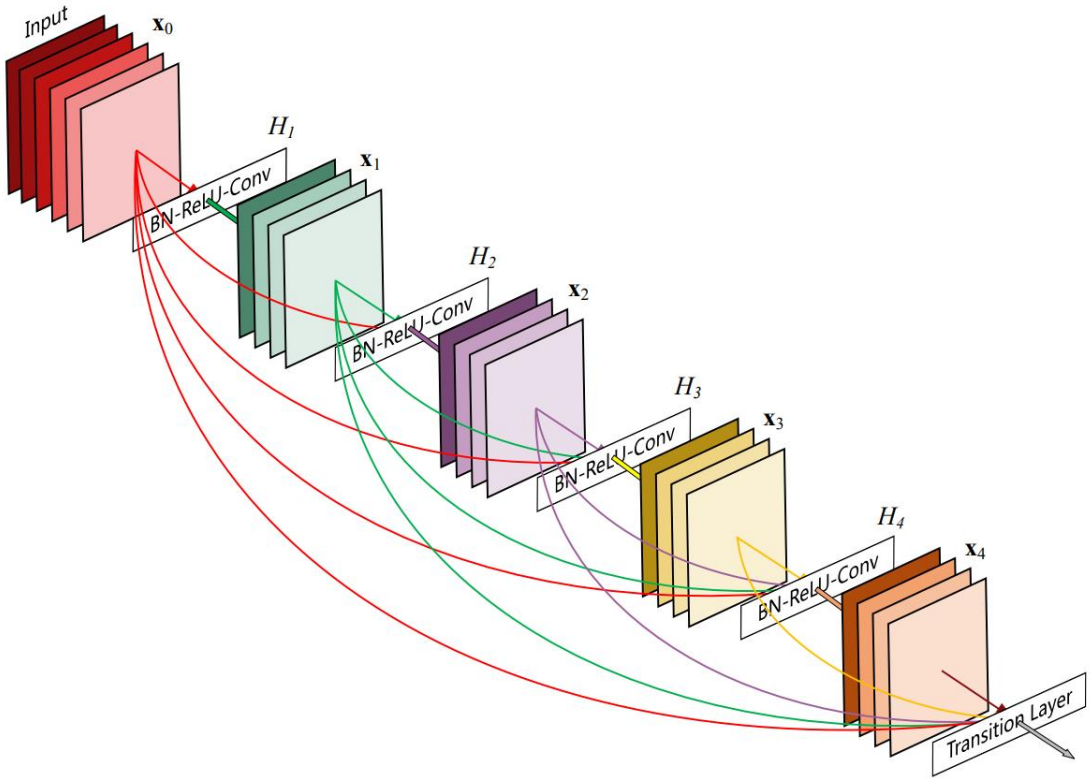


Figure 3. DenseNet-201 Architecture

4.2.4. Inception Modules

Inception architectures are characterized by their Inception modules, which are composed of multiple parallel convolutional filters of different sizes. These modules are crafted to extract features at multiple scales within a single layer. As a result, the network

can process information at varying resolutions effectively. This approach avoids the need for large, computationally expensive filters, ensuring greater efficiency. Utilizing smaller filters decreases both the number of parameters and the computational load, enhancing the network's efficiency. Inception architectures have proven effective across numerous computer vision tasks, including applications like image classification and object detection. They represent an approach to deep learning that balances accuracy and computational efficiency, making them suitable for real-world applications where both factors are important [17], [70].

GoogleNet, also known as Inception v1, was introduced by Szegedy et al. [17], is a 22-layer CNN network that utilizes the inception method presented during the ILSVRC2014 competition. In crafting this CNN network, the authors acknowledged that the objects in the images might differ in size, necessitating kernels of different dimensions to accommodate the distribution of information. Therefore, in the inception method, different-sized kernels can be used at similar CNN network levels. GoogleNet comprises 22 deep convolutional layers. If layers with pooling are also considered in terms of depth, it would have 27 deep layers. Following the initial two convolutional and maximum pooling layers, GoogleNet uses 9 inception modules. In the näive inception module, three different-sized convolutional layers are applied to the output of the previous layer or module, i.e., the feature matrix. The kernel sizes used in these convolutional layers are 1 x 1, 3 x 3, and 5 x 5. Additionally, a 3 x 3 maximum pooling layer is used. In the näive inception module, the outputs from each convolutional and max pooling layer are merged and then passed on to the subsequent inception module. An example of the inception module can be seen in Figure 4.

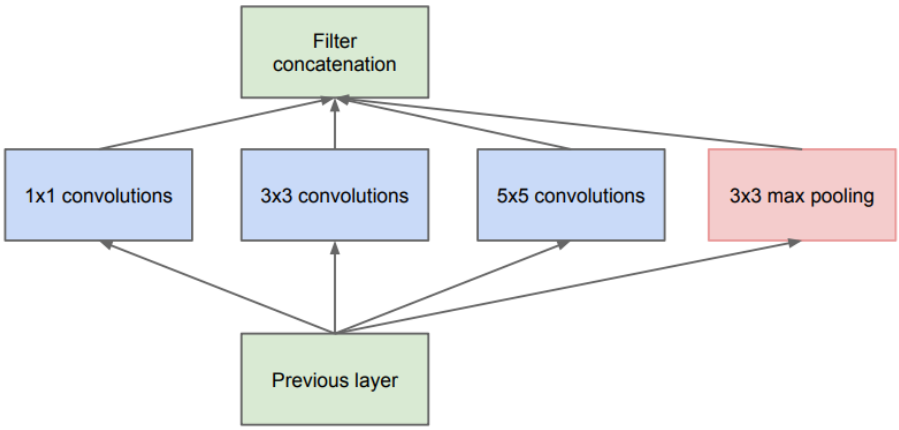


Figure 4. Inception module, naïve version

For the purpose of reducing the additional computational load in pure inception module, a dimension-reduced inception module is proposed. In the dimension-reduced inception module, a 1 x 1 convolutional layer is applied before the 3 x 3 and 5 x 5 convolutional layers. Additionally, after the 3 x 3 maximum pooling layer, a 1 x 1 convolutional layer is applied. The goal of incorporating dimension-reduced inception modules is to decrease the number of parameters and computational demands while still effectively capturing key features. To achieve this, 1 x 1 convolutional layers are employed to reduce the dimensionality of the feature maps before applying the more computationally demanding 3 x 3 and 5 x 5 convolutional layers. This results in a more computationally efficient architecture while still benefiting from the feature diversity provided by the inception architecture. Dimension-reduced inception module is proposed in Figure 5.

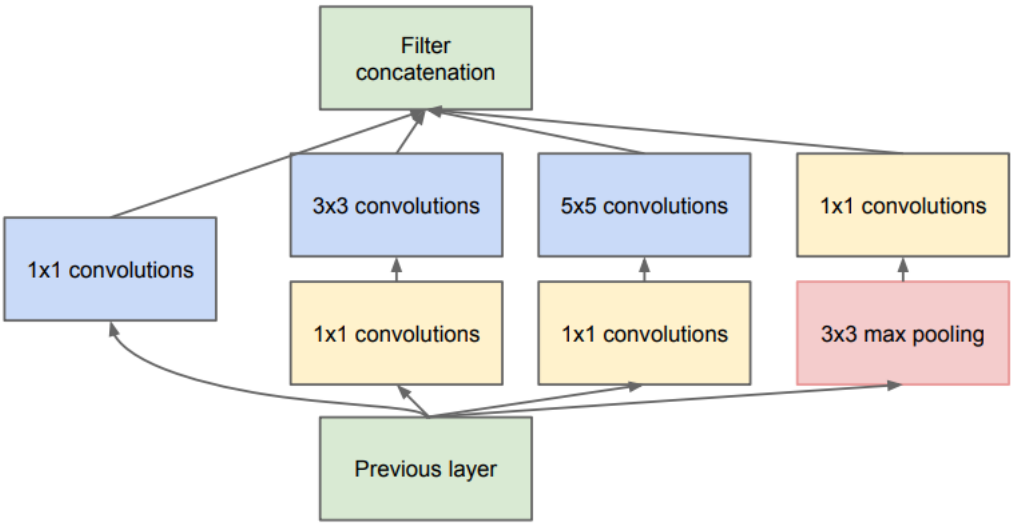


Figure 5. Inception module incorporating dimension reduction

Inception v2 is an improved version of the GoogleNet (Inception v1) CNN architecture presented by Szegedy et al. [71] and is an optimized deep learning network that serves as Inception v2 simplifies the complexity found in the GoogleNet architecture, making the convolutional neural network more adaptable to various conditions. Several general principles and optimization ideas were introduced alongside Inception v2 to enable the scalability of the convolutional network such as avoiding representational bottlenecks, balancing the network's size and layers, and easier processing of higher-dimensional representations within the local network. In GoogleNet, 5 x 5 convolutional layers were used in some cases to significantly reduce input dimensions, which could lead to a reduction in the accuracy of the CNN. In GoogleNet, 5x5-sized filters were used, whereas in the Inception v2 model, they opted for consecutive 3x3-sized filters (see Figure 6). This change helped reduce the computational load and resulted in an improved model.

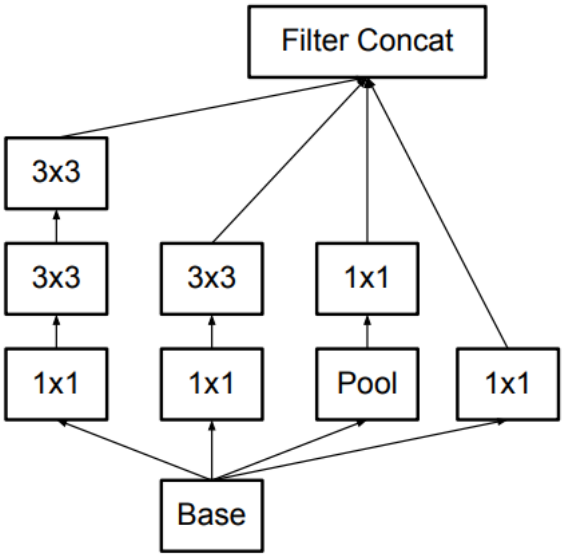


Figure 6. Two times 3x3 convolution is in place of 5x5 convolution

Inception v2 processes inputs differently compared to a straightforward n x n operation. Instead of directly applying a convolution to an n x n input, it consecutively applies 1 x n and n x 1 convolutions, i.e., in the 3x3 convolution layer of the Inception v2 network, the input tensor is first convolved with a 1 x 3 kernel, and then a 3 x 1 kernel is applied to the output, resulting in the final output (see Figure 7). This approach reduces the computational cost by approximately 33% [71].

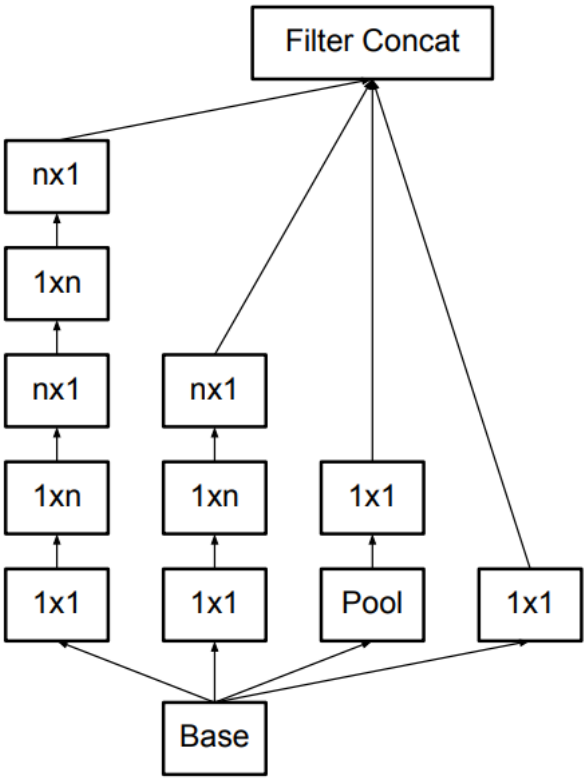


Figure 7. Using 1 x n and n x 1 kernels instead of an n x n kernel in certain modules

Traditionally, convolutional networks have relied on pooling operations to downsize feature maps. The network filters' dimensions are increased beforehand for the purpose of avoid a bottleneck in data representation. This expansion occurs before maximum or average pooling is applied, as illustrated in Figure 8.

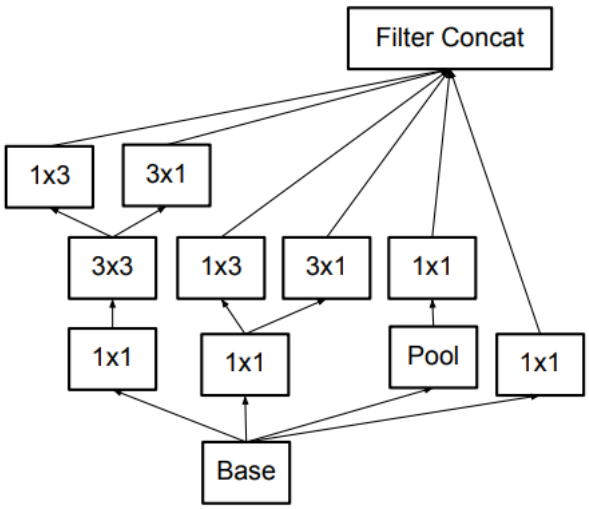


Figure 8. Inception modules that increase the number of filter bank outputs

Architecture of Inception v2 is summarized in Table 5.

Table 5. Inception v2 Architecture

Type	Patch size/stride or remarks	Input size
Convolutional	3×3/2	299×299×3
Convolutional	3×3/1	149×149×32
Convolutional padded	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Convolutional	3×3/1	73×73×64
Convolutional	3×3/2	71×71×80
Convolutional	3×3/1	35×35×192
3 x Inception	As in Figure 4	35×35×288
5 x Inception	As in Figure 5	17×17×768
2 x Inception	As in Figure 6	8×8×1280
Pool	8 x 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
Softmax	Classifier	1 × 1 × 1000

Inception v3 is considered as one of the most popular and advanced architectures for image classification tasks. It is an improvement and optimization over its predecessors, v1 and v2, offering a deeper convolutional architecture while maintaining high efficiency, faster processing, and significantly lower error rates. This model consists of 42 layers and uses an approach that combines convolutional filters in various dimensions into a single filter. It has undergone modifications and updates in Inception modules, the auxiliary classifier, and the grid system. As a result, Inception V3 maintains high efficiency with reduced computational cost, resulting in fewer parameters to train and lower computational complexity [71]. A schematic representation of the Inception V3 model is shown in Figure 9.

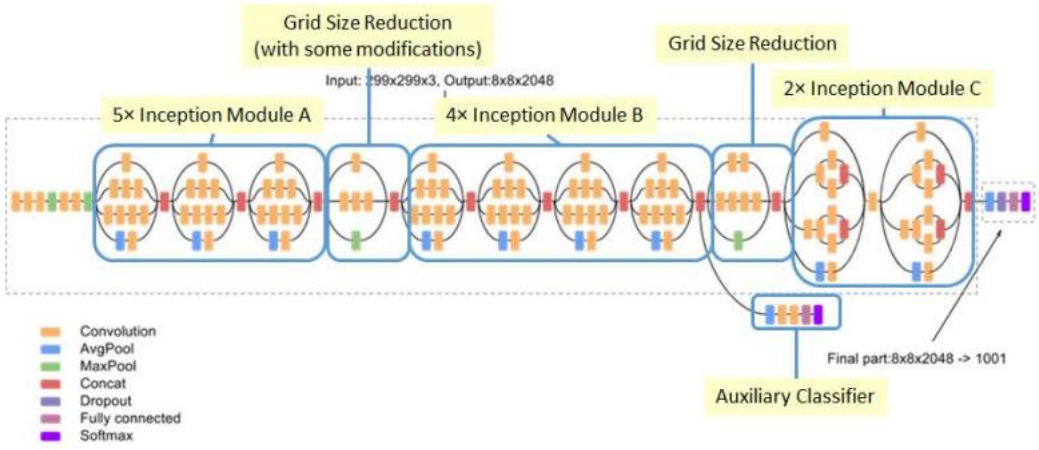


Figure 9. The diagrammatic overview of the Inception V3 model

4.2.5. Swin Transformer

Swin Transformer was introduced in 2021, addressing the substantial differences in the scale of image components and the increased pixel resolution within images. The fundamental premise of this method is based on utilizing Sliding Window-based Self-Attention. The traditional module has a limitation, connection between windows are lacking. This lack of interwindow connection reduces the module's ability to effectively model and understand the overall structure, or patterns of the data. The Swin Transformer framework introduces a technique that splits the input image into smaller, overlapping windows. This approach enables the model the model to manage different areas of the image independently, making it more effective at processing distinct image parts. The sliding window partitioning technique improves the model's capacity to capture both fine-grained and broader features in high-resolution images and increases efficiency by limiting self-attention calculations to distinct, non-overlapping local regions. The technique is depicted in Figure 10.

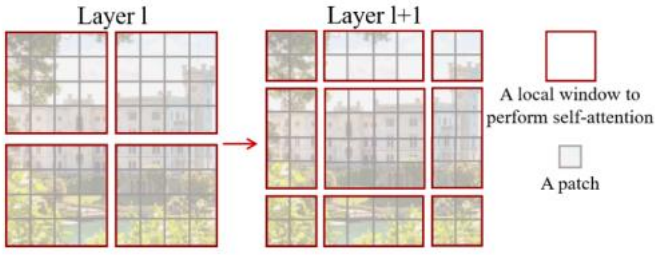


Figure 10. Sliding Window Approach in Swin Transformer [45]

4.2.6. Vision Transformer

ViT presents a DL framework [40] that leverages the Transformer model for processing and extracting important information from images. The design consists of

multiple layers, including patch embedding, positional encoding, transformer encoder layers, and a classification head. Each of these components is essential for the comprehensive analysis of the input image. The structure is illustrated in Figure 11.

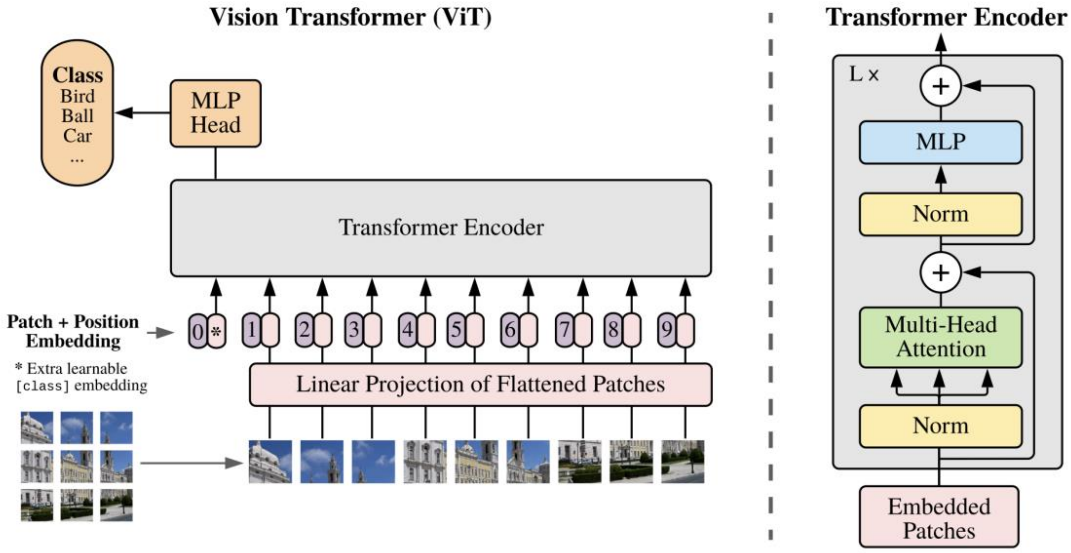


Figure 11. Overview of the Vision Transformer Architecture

The patch embedding layer receives the image patch grid as input and transforms each one into a smaller embedding vector utilizing a learnable linear projection. The result is a series of patch embeddings that serve as input tokens for the next layers.

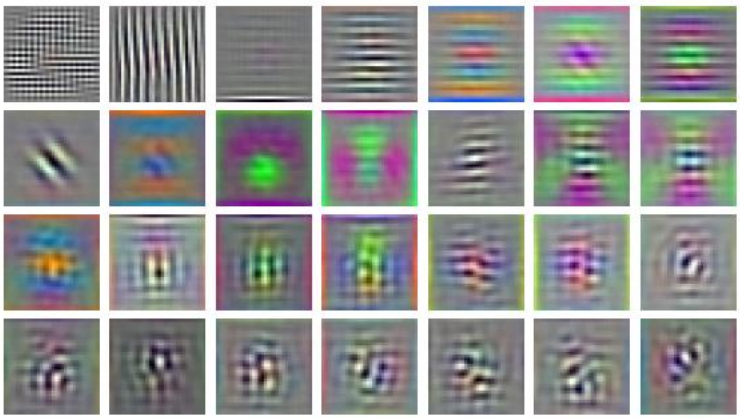


Figure 12. RGB embedding filters of ViT-L/32 model

For the model to understand spatial positions of the patches, positional encoding injects position information into the patch embeddings. It helps the model to capture spatial relationships and differentiate between various positions in the image.

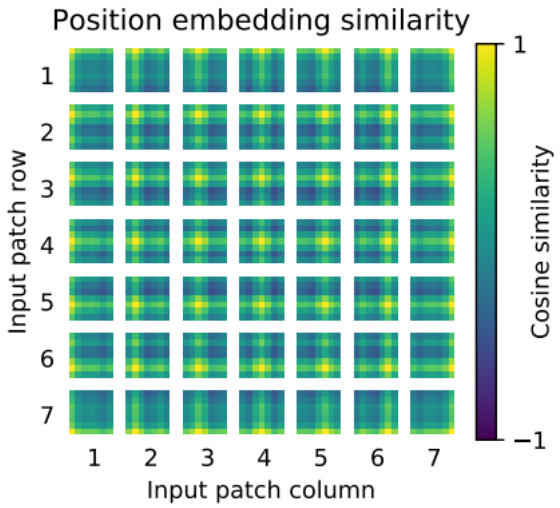


Figure 13. The similarity of location embeddings of ViT-L/32 model

The core structure of the model includes several Transformer encoder layers. Each of these layers is made up of two sub-layers: MHSA and FFNN. The MHSA sub-layer enables each patch to focus on every other patch, thereby identifying global dependencies and connections. Within the input sequence, the self-attention mechanism detects relationships between various patches. For each patch embedding, it computes a sum of all patch embeddings where each patch's contribution to the sum is scaled by its relevance to the current patch. This mechanism permits the model to focus on crucial patches while accounting for local and global contexts.

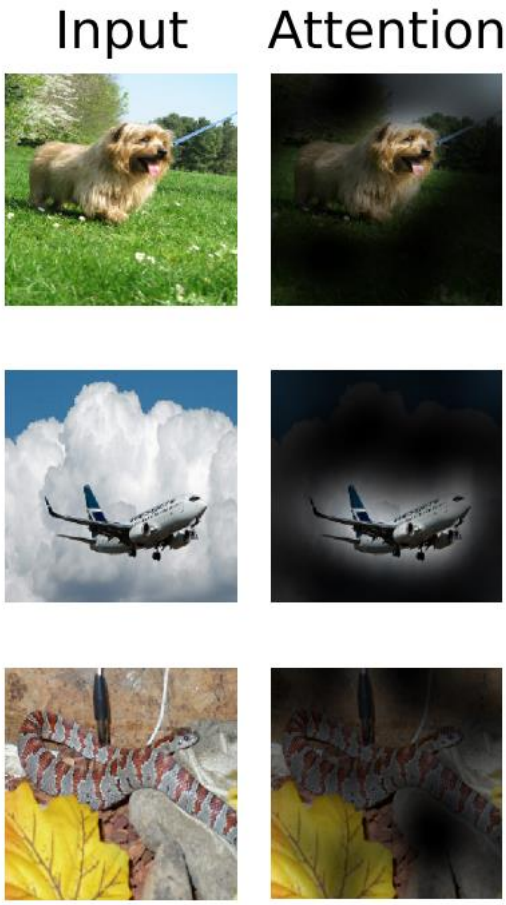


Figure 14. Representative Examples of Attention

After self-attention is employed, the output generated by each patch's self-attention is passed through an FFN. Typically, a fully connected layer succeeded by ReLU is included in this network. The purpose of this design is to introduce non-linearity and allow the model to learn intricate relationships between patches.

Classification head is the final layer and responsible of predicting the label of the image. ViT utilizes the Transformer architecture and the capabilities of self-attention to identify both local and global features within the image patches.

4.2.7. ConvNeXt

The ConvNext is a further advancement of the ResNet model which uses the design of Swin excluding attention-based modules. While ResNet uses the ReLU function and batch normalization, ConvNeXt adopts the GELU as an activation function, and layer normalization. When compared to the ResNet, ConvNeXt has fewer activation functions.

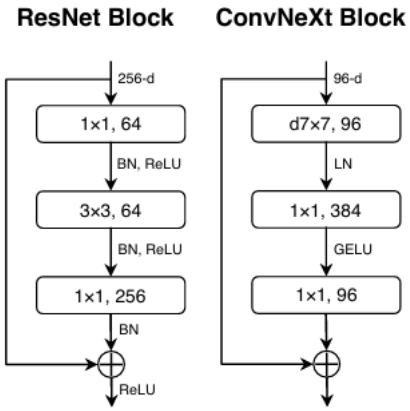


Figure 15. Block designs of ResNet and ConvNext [72]

In particular, each block is streamlined to include only a single GELU. Furthermore, the quantity of normalization functions are reduced, each block is left to contain only one layer, “layer normalization”. Patchify layer in Swin Transformer is adapted using a 4x4 convolutional layer which is non-overlapping with stride 4. Introduction of depthwise convolution along with a larger kernel size in each block is another improvement of ConvNext model. Depthwise convolution is a form of grouped convolutions which works on a per-channel basis. Either spatial information or channel information can be separated, never both.

4.2.8. MaxViT

MaxViT is a hybrid model that merges the advantages of enhanced convolutions and attention mechanisms. It consists of two convolutional blocks succeeded by a series of MaxViT blocks. A single MaxViT block initially has a MBConv block which is also called as inverted residual block. Narrow → wide → narrow approach is followed as a structure approach. When compared to a standard residual block [50], this structure approach greatly reduced the quantity of parameters. SE module is located between 3x3 depthwise convolution and 1x1 final convolution to model interdependencies among channels. After MBConv block, Block Attention is used to capture local features. It is followed by Grid Attention to capture global features. In each MaxViT block, the quantity of spatial filters are different from each other. An average pooling layer comes before the Multi-Layer Perceptron Network (MLP) to classify the input. MaxViT architecture is given in Figure 16.

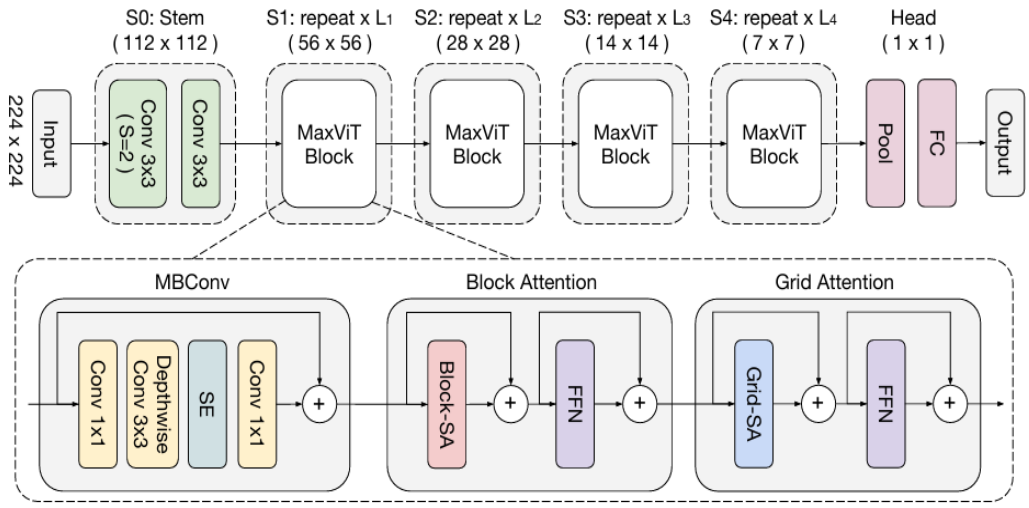


Figure 16. MaxViT architecture [42]

The relative attention is described as:

$$RelAttention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}} + B\right)V \tag{4.1}$$

Where Q, K, V ∈ $\mathbb{R}^{(HxW)xC}$ are the query, key, and value matrices and d is the hidden dimension. Attention weights are controlled by a learned fixed location-aware matrix B and normalized input-adaptive attention.

The relative attention function in Equation (4.1) is adhering to the convention for 1D input sequences. This approach interprets the second last dimension of an input $((\ldots, L, C)$ as the spatial axis, where L and C denote sequence length and channels. Block $(.)$ operator, parameter P, partitions the input image/feature $x \in \mathbb{R}^{(HxW)xC}$ into non-overlapping blocks, each of size $P\ x\ P$. After partioning into windows, the block dimensions are arranged along the spatial dimension.

$$Block : (H,W,C) \rightarrow \left(\frac{H}{P}\ x\ P, \frac{W}{P}\ x\ P, C\right) \rightarrow \left(\frac{HW}{P^2}, P^2, C\right) \tag{4.2}$$

The $Unblock\ (.)$ operation reverses the block partition process. Similarly, the $Grid\ (.)$ operator with parameter G divides the input feature into a uniform $G\ x\ G$ grid where each cell has an adaptive size $\frac{H}{G}\ x\ \frac{W}{G}$. Unlike the Block operator, the Grid operator demands

an additional transpose to place the grid dimension in the expected spatial axis (i.e., the second-to-last axis):

$$Grid : (H, W, C) \rightarrow \left(G\ x\ \frac{H}{G}, G\ x\ \frac{W}{G}, C\right) \rightarrow \left(G^2, \frac{HW}{G^2}, C\right) \rightarrow \left(\frac{HW}{G^2}, G^2, C\right) \quad (4.3)$$

The inverse operation, $Ungrid\ (.)$ reverses the gridded input back to the standard 2D feature format. With this groundwork, multi-axis attention module is explained. For a given input tensor $x\ \in\ \mathbb{R}^{(HxW)xC}$, the local Block Attention can be formulated as:

$$x\ \leftarrow x + Unblock\left(RelAttention\left(Block\left(LN(x)\right)\right)\right), \quad (4.4)$$

$$x\ \leftarrow x + MLP(LN(x)), \quad (4.5)$$

The global, dilated Grid Attention can be formulated as:

$$x\ \leftarrow x + Ungrid\left(RelAttention\left(Grid\left(LN(x)\right)\right)\right), \quad (4.6)$$

$$x\ \leftarrow x + MLP(LN(x)), \quad (4.7)$$

In these equations, the QKV input format in the RelAttention is omitted for simplicity. Layer normalization and multi-layer perceptron are abbreviated as LN and MLP, respectively.

4.3. Activation Function

Neural networks are built from individual units called neurons, where each unit calculates a sum of inputs multiplied by specific weights and then applies a function to determine its output. Excluding an activation function from the network might lead the network to merely perform linear transformations. The range of the neuron's output is controlled by different activation functions. An appropriate activation function greatly enhances the performance of a network. Some of the well-known and common activation functions are ReLU, tanh and sigmoid.

Even if ReLU helps to mitigate issues like vanishing gradient problem or computes faster than both sigmoid or tanh functions [73], neither ReLU nor tanh are applicable for multi-label image classification tasks. Output values from ReLU activation function are in

the range [0, ∞) where the range is in between [-1, 1] for tanh. Multi-label classification requires the outputs to represent probabilities for each label and should be in the range [0, 1]. Equations for both functions are given in Eq. 4.8 and 4.9.

$$ReLU(x) = \max(0, x) \tag{4.8}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.9}$$

Sigmoid function is used in this thesis as an activation function. The reason for it to be used is not limited with its output range ([0, 1]), but also it provides smooth gradients that helps with backpropagation during training phase. This ensures that the learning algorithm can effectively update the weights. Sigmoid outputs allow high probabilities for all labels. The sigmoid function is calculated as in Eq. 4.10.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.10}$$

4.4. Loss Function

Loss functions play essential role in training neural networks by measuring the inconsistency between the prediction and true labels. It determines the performance of an algorithm on a dataset. Reducing the loss function is a key goal when using a machine learning algorithm to train a model. There are many loss functions to assess the error in a neural network.

Cross Entropy Loss is one of the most traditional loss function in image classification applications. Even if it is widely used, cross-entropy loss is not appropriate in multi-label classification tasks. A single class per sample is expected which represents the true label. But in MLIC, each class can be annotated with different classes concurrently. Additionally, it is designed to work with a single probability distribution among classes as given in Eq. 4.11 where p_i is the predicted probability for the class i (softmax output) and t_i is the true label.

$$\text{Cross} - \text{Entropy Loss} = -\sum_{i=1}^{m} t_i \log(p_i) \tag{4.11}$$

A proper loss function selection is important for a specific task which is multi-label image classification for this thesis. Two functions are applied as loss functions to the different networks, Binary Cross-Entropy With Logits Loss (BCEWithLogitsLoss) [74] and Asymmetric Loss (ASL) [75].

A given problem with multiple labels is divided into distinct binary problems for each label by BCEWithLogitsLoss which makes it possible to assign multiple labels to each item. The sigmoid function produces separate real-valued outputs for every input. Thus, the actual output is estimated. Sigmoid allows high probabilities for all classes, unlike Softmax which assigns high probability to high value. The logits represent the raw outputs before Sigmoid is applied. Formula for BCEWithLogitsLoss is given in Eq. 4.12 where p_i' is fully connected layer's logits (Sigmoid function is applied) and p_i is the true label.

$$BCEWithLogitsLoss = -\frac{1}{n}\sum_{i=1}^{n} p_i \log(p_i') + (1 - p_i)\log(1 - (p_i'))] \quad (4.12)$$

ASL is one of the variations of binary cross-entropy loss. They are generally combined by a sigmoid function (Eq. 4.10) for the purpose of converting the model outputs to probabilities. ASL consists of two complementary asymmetric mechanisms, that operate in distinct ways on positive and negative samples, enabling direct control. ASL enables for the selective increase in weight of minority negatives and meanwhile maintains the original weighting for common positives, biasing frequent classes is dismissed by this. Re-balancing is also enabled by ASL, unlike batch-dependent schemes such as distribution-balanced loss. The ASL formula is expressed as in Eq. 4.13.

$$ASL = -\sum_{n=1}^{N} [p_i \log(p_i') + (1 - p_i)\log(1 - p_i')]\,(p_i(1 - p_i') + (1 - p_i)p_i')^{(\gamma^+ y) + (\gamma^- (1-y))} \quad (4.13)$$

Eq 4.14. is a more understandable and simplified version of Eq. 4.13.

$$ASL = -\sum_{n=1}^{N} (p_i(1 - p_i')^{\gamma^+} \log(p_i') + (1 - p_i)(p_i'^{\gamma^-})(\log(1 - p_i'))) \quad (4.14)$$

γ^+ and γ^- are two pre-defined hyperparameters in the equation above which are set as 1 and 2, respectively, in this thesis. The final loss function is given in Eq. 4.15.

$$ASL = -\sum_{n=1}^{N} (p_i(1 - p_i')\log(p_i') + (1 - p_i)(p_i'^{2})(\log(1 - p_i'))) \quad (4.15)$$

If $\gamma^+$and γ^- are both set to be 0 in Eq. 4.14 , BCEWithLogitsLoss can be calculated as well (See Eq. 4.12).

4.5. Evaluation Metrics

There are four cases which a classifying model can give as an output: true positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). These metrics can be used to assess a model's precision, recall and F1-score values. When a model gives positive outputs, the precision shows its reliability. When a model correctly classifies positive data points, that is the recall. Harmonic mean of both recall and precision is equal to F1-Score.

Various evaluation metrics are used for the multi-label image classification as determination of this thesis' success. The classification report tool is used from sklearn.metrics library. The evaluation metrics are calculated in the Equations below.

- Precision,
- Recall,
- F1-Score,
- Macro Average,
- Micro Average,
- Weighted Average,
- Mean Average Precision (mAP),

Precision is the ratio of the number of correct positive samples predicted by the model to the total number of positive samples.

$$Precision = \frac{TP}{TP+FP} \tag{4.16}$$

Recall is the ratio of the number of true positive samples to the sum of true positive and false negative samples. It reveals how many positive data the model's prediction misses.

$$Recall = \frac{TP}{TP+FN} \tag{4.17}$$

One of the most popular evaluation metrics in the machine learning applications is the F1-Score. As shown below, it is the harmonic ratio of Precision and Recall.

$$F1 - Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4.18}$$

Let $B(TP_j,\ FP_j,\ TN_j,\ FN_j)$ be a specific binary classification metric, where $B \in \{Precision, Recall, F1 - Score\}$. [76]

$$B_{macro} = \frac{1}{q}\sum_{j=1}^{q} B\left(TP_j, FP_j, TN_j, FN_j\right) \tag{4.19}$$

$$B_{micro} = B\left(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} TN_j, \sum_{j=1}^{q} FN_j\right) \tag{4.20}$$

$$B_{weighted} = B(\sum_{j=1}^{q} TP_j.\frac{support_j}{p}, \sum_{j=1}^{q} FP_j.\frac{support_j}{p}, \sum_{j=1}^{q} TN_j.\frac{support_j}{p}, \sum_{j=1}^{q} FN_j.\frac{support_j}{p}) \tag{4.21}$$

Mean Average Precision (mAP) measures the overall performance across all classes. In other terms, it is the mean of the average precisions (AP) of all classes where AP is the area under the precision-recall curve. mAP can be calculated as shown in Eq. 4.22 where L denotes the number of classes.

$$mAP = \frac{1}{L}\sum_{i=1}^{L} P(i).R(i) \tag{4.22}$$

Precision-recall curve is used in this thesis due to the reason that ROC-curve's estimation might be inadequate as long as the positive class is substantially smaller [77]. Precision-recall curve assesses how well a model distinguishes between positive and negative classes. The curve offers valuable insights into the balance between precision and recall at various thresholds. To assess the algorithms' performance, the area under the precision-recall curve is computed. This area serves as an important metric for evaluating

the quality of the models. The ideal curve should be close to the top-right corner of the plot, where both precision and recall are 1.

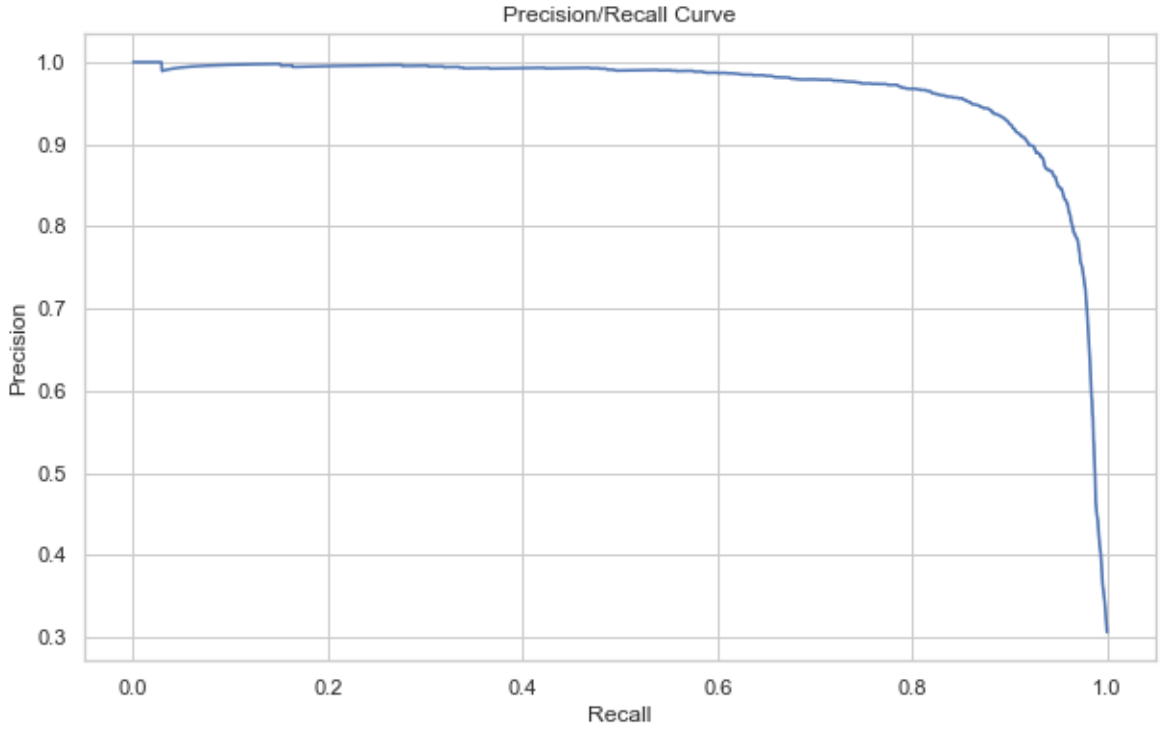


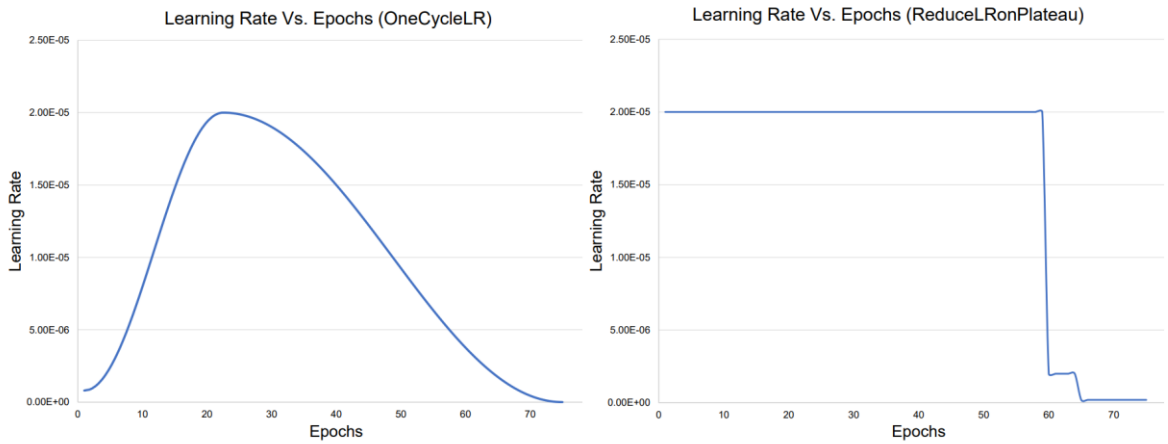
Figure 17. An example of Precision-Recall Curve

4.6. Learning Rate and Scheduler

The learning rate influences how quickly, steadily, and effectively a model learns, making it a critical component in model training. Tuning the learning rate in an appropriate way is essential for successful model training. A learning rate scheduler, or simply scheduler, is a method that modifies the learning rate throughout the training process based on a set plan. Smith et al. [78] introduced the OneCycleLR learning rate policy, which aims to fine-tune the learning rate to prevent the model from settling at a local minimum that is only beneficial for the training set. This approach aims to adjust the learning rate dynamically throughout training. Compared to other scheduling methods, OneCycleLR achieves super-convergence guiding the model toward an optimal point for the loss function, improving both training and validation outcomes. On the other hand, just like OneCycleLR, learning rate is dynamically updated during training by ReduceLROnPlateau [79] to ensure the model keeps improving its performance even if plateau is reached.

Both schedulers are provided by the torch.optim library as OneCycleLR and the ReduceLROnPlateau. Minimum 2 functions are needed for OneCycleLR to function: the peak learning rate value (*max_lr*) and the number of epochs (*epochs*). At least 4 functions are required for ReduceLROnPlateau:

- Initial learning rate: It comes from the choosen optimizer.
- Validation loss: To evaluate the model accuracy.
- *patience*: The allowed number of consecutive epochs without improvement before the learning rate is reduced.
- *factor:* It decreases the lr by a given factor.



(a) `OneCycleLR` (b) `ReduceLROnPlateau`

Figure 18. Overview of OneCyleLR (a) and ReduceLROnPlateaue (b) schedulers [80]

In this thesis, Adam [81] is used for the optimization where the learning rate is 0.0001 with a weight decay of 0.0001. For ReduceLROnPlateaue, patience is set to 2 to track the validation loss value and *factor* is set to 0.1 (reducing the learning rate to $lr * factor$). For OneCycleLR, maximum learning rate is set to $4 * 10^{-4}$.

5. DISCUSSION AND RESULTS

This chapter provides an in-depth examination of the findings derived from the MLIC experiments. The central objective of this study was to achieve best classification results for Multi-label AID dataset which can enrich the multi-label image classification tasks, yet it remains relatively unknown in the broader research community currently. The limited conducted research on this specific dataset mainly focused on utilizing both visual and spatial information by combining CNN, mostly ResNet50 [66], [67], [68], and inherent correlation of labels [69]. Even if the results are promising from the prior research, the motivation behind this thesis is to increase the value of the evaluation metrics and contribute in the research community by classifying multi-label aerial images which plays a vital role in remote sensing applications. Through an extensive series of experiments, several SOTA models are evaluated with the support by different activation functions, loss functions and hyperparameters.

During the literature review, it is observed that numerous neural networks such as CNN, DCNN, fused networks, attention modules and vision transformers, achieved significant results. The contributions of these applications are not limited to achieving promising outcomes in multi-label image classification problems, they also shed light on this thesis. The evaluation of the data and results obtained from all these studies has demonstrated that the MaxViT algorithm is highly successful in numerous applications. However, it has been noted that such a successful hybrid network has not yet been applied to the multi-label AID dataset.

MaxViT is a hybrid model that initially consists of CNN and transformer network, combining global and local features with a simple and scalable design while maintaining computational efficiency. The comparison of MaxViT with the other well-known models is given in Table 6.

Table 6. Benchmarking MaxViT against different CNN and Transformer Models

Model	Type	Key Features	MaxViT Comparison
AlexNet	CNN	Simple architecture, foundational deep learning model	MaxViT provides advanced features, better scalability

VGG16	CNN	Deep architecture with 16 layers	MaxViT offers better efficiency, global-local interactions
DenseNet-201	CNN	Dense connections, efficient parameter usage	MaxViT is more scalable, efficient with global interactions
InceptionV3	CNN	Factorized convolutions, multi-scale processing	MaxViT simplifies design, maintains high performance
ConvNeXt	Modern CNN	Transformer-inspired designs, competitive performance	MaxViT integrates convolution and attention mechanisms
ViT	Transformer	Image patches as sequences, transformer-based	MaxViT combines convolution and attention for scalability
SwinT	Transformer	Shifted window attention, hierarchical architecture	MaxViT offers linear complexity global-local interactions

The MaxViT-T model demonstrates a compelling advantage by combining CNN and transformers with 31M parameters and 5.6G FLOPs. MaxViT-T achieves 83.6% accuracy on the ImageNet-1K dataset, surpassing AlexNet and VGG16 with 60M and 138M parameters as well as 63.3% and 71.5% accuracy respectively. Even more advanced models like DenseNet-201 and Inceptionv3 do not match MaxViT's performance with accuracies of 77.2% and 77.9%. Transformers such as ViT and SwinT require extensive pre-training and struggle with global interaction capacities. MaxViT-L achieves superior results with 85.2% accuracy with 212M parameters and 43.9G FLOPs. Due to the computational complexity of MaxViT-L, MaxViT-T is chosen as a main network in this thesis even if it is not as successful as MaxViT-L. Features and performances of aforementioned networks and MaxViT are given in Table 7.

Table 7. Features and performances of the models

Model	Parameters	FLOPs	ImageNet-1K Accuracy
AlexNet	60M	720M	63.3%
VGG16	138M	15.3G	71.5%
DenseNet-201	20M	4.3G	77.2%
InceptionV3	24M	5.7G	77.9%
ViT	86M	55.4G	77.9%

SwinT	50M	8.7G	83.0%
ConvNeXt	50M	8.7G	83.1%
MaxViT-T	31M	5.6G	83.6%
MaxViT-L	212M	43.9G	85.2%

For the experiments: Pytorch is used as the primary deep learning framework on a personal computer with NVIDIA Geforce RTX 3070 Ti graphic card with 16 GB of memory.

Due to the scant quantity of the Multi-label AID images, it is more likely for the models to overfit. For the purpose of preventing overfitting, not only the data is augmented, but also patience parameter is selected as 5 to keep the models with the best validation accuracy under 20 epochs. In addition, one Dropout layer with 20% probability is added just before the very last layer, fully connected layer, for each model.

Transfer learning is employed using pre-trained weights for all the models. These weights were obtained from models pre-trained on ImageNet. Models that are pre-trained consistently outperform those trained from scratch [41].

The math behind MLIC is different than SLIC. Softmax is one of the loss functions that enforces mutual exclusivity among classes, and it is unsuitable for multi-label classification where multiple classes can be assigned independently. Sigmoid is used as activation function in multi-label classification tasks since it allows each class probability to be calculated independently. Sigmoid is combined with binary cross-entropy loss, one of the most widely used loss functions, where each label is handled as an independent binary classification problem rather than using cross-entropy loss. For better understanding, Table 8 and Figure 19 shows the results where *softmax & cross entropy loss* and *sigmoid & binary cross entropy loss with logits* is applied on Inceptionv3 module. Cross Entropy Loss, when used together with softmax forces the sum of probabilities to 1, and it is not suitable when multiple labels can be true simultaneously. This assumption leads the model to suboptimal performance. To support this proposal, the same functions have also been tested on the MaxViT algorithm, given in Table 9 and Figure 20. Based on these comparisons, *sigmoid* and *binary cross entropy loss with logits* achieve much better results for MLIC than the *softmax* and *cross entropy loss* that are commonly used in SLIC tasks.

Table 8. Evaluation metrics using different activation and loss functions for Inceptionv3

Inceptionv3						
Labels	**Softmax + Cross Entropy Loss**			**Sigmoid + BCEWithLogitsLoss**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
airplane	0,54	1	0,7	1	0,6	0,75
bare-soil	0,56	0,98	0,71	0,8	0,73	0,76
buildings	0,87	0,91	0,89	0,93	0,98	0,95
cars	0,89	0,08	0,15	0,9	0,97	0,94
chaparral	0,25	0,24	0,25	0,75	0,08	0,15
court	0,67	0,03	0,05	0,88	0,39	0,54
dock	0	0	0	0,74	0,74	0,74
field	0,67	0,46	0,55	0,77	0,62	0,69
grass	0,69	0,16	0,26	0,95	0,94	0,94
mobile-home	N/A	N/A	N/A	N/A	N/A	N/A
pavement	1	0	0	0,95	0,99	0,97
sand	0,65	0,79	0,71	0,85	0,87	0,86
sea	0	0	0	0,95	0,93	0,94
ship	0	0	0	0,62	0,85	0,71
tanks	0	0	0	0,95	0,86	0,9
trees	0,81	0,05	0,09	0,96	0,93	0,94
water	0	0	0	0,9	0,62	0,73
micro avg	0,68	0,29	0,41	0,91	0,88	0,89
macro avg	0,45	0,28	0,26	0,82	0,71	0,74
weighted avg	0,71	0,29	0,29	0,91	0,88	0,88

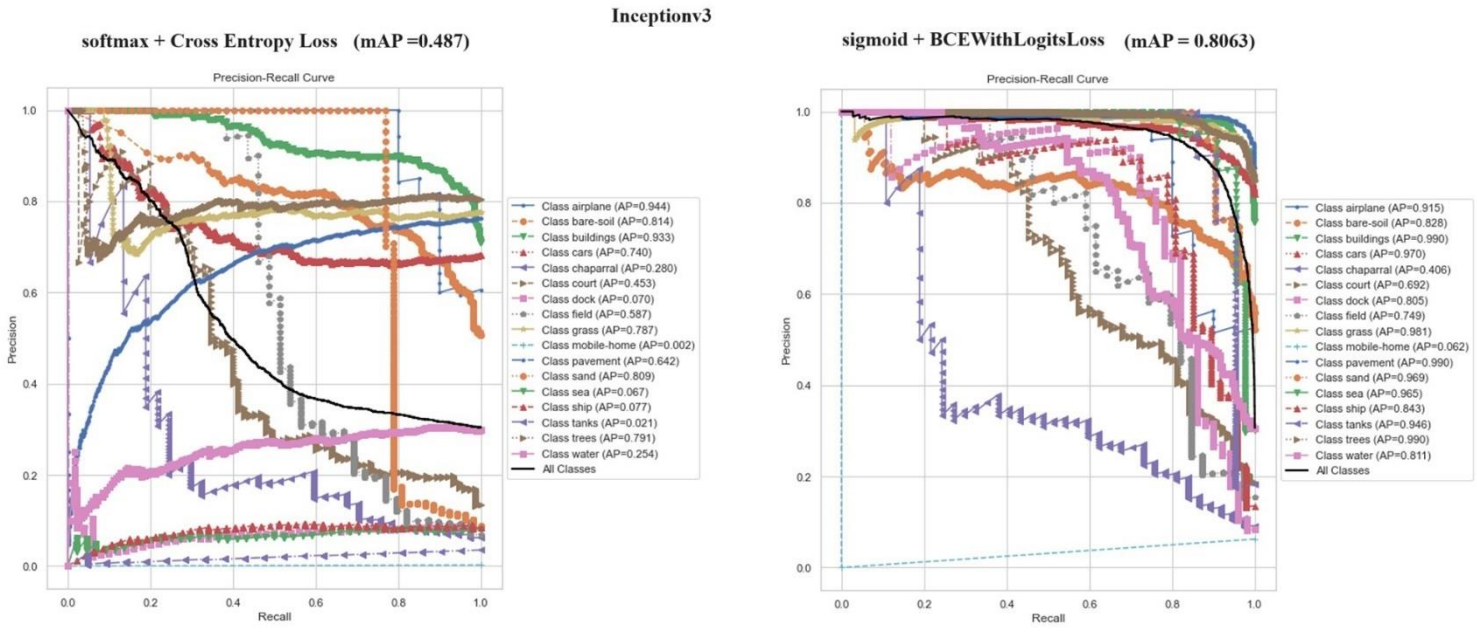


Figure 19. Comparison of using different activation and loss functions for Inceptionv3

Table 9. Evaluation metrics using different activation and loss functions for MaxViT

MaxViT						
Labels	**Softmax + Cross Entropy Loss**			**Sigmoid + BCEWithLogitsLoss**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
airplane	0,61	1	0,75	0,92	0,55	0,69
bare-soil	0,67	0,96	0,79	0,78	0,79	0,78
buildings	0,88	0,66	0,75	0,97	0,97	0,97
cars	0,83	0,1	0,18	0,95	0,91	0,93
chaparral	0,25	0,22	0,23	0,35	0,3	0,32
court	0	0	0	0,54	0,83	0,66
dock	0	0	0	0,9	0,52	0,66
field	0,57	0,51	0,54	0,89	0,41	0,56
grass	0,69	0,14	0,24	0,94	0,96	0,95
mobile-home	N/A	N/A	N/A	N/A	N/A	N/A
pavement	0,07	0	0	0,97	0,97	0,97
sand	0,71	0,79	0,75	0,98	0,77	0,86
sea	0	0	0	0,95	0,91	0,93
ship	0	0	0	0,9	0,57	0,7
tanks	0	0	0	0,91	1	0,95
trees	0,64	0,04	0,08	0,96	0,94	0,95
water	0	0	0	0,92	0,61	0,73
micro avg	0,71	0,25	0,37	0,91	0,88	0,9
macro avg	0,35	0,26	0,25	0,81	0,71	0,74
weighted avg	0,53	0,25	0,28	0,92	0,88	0,89

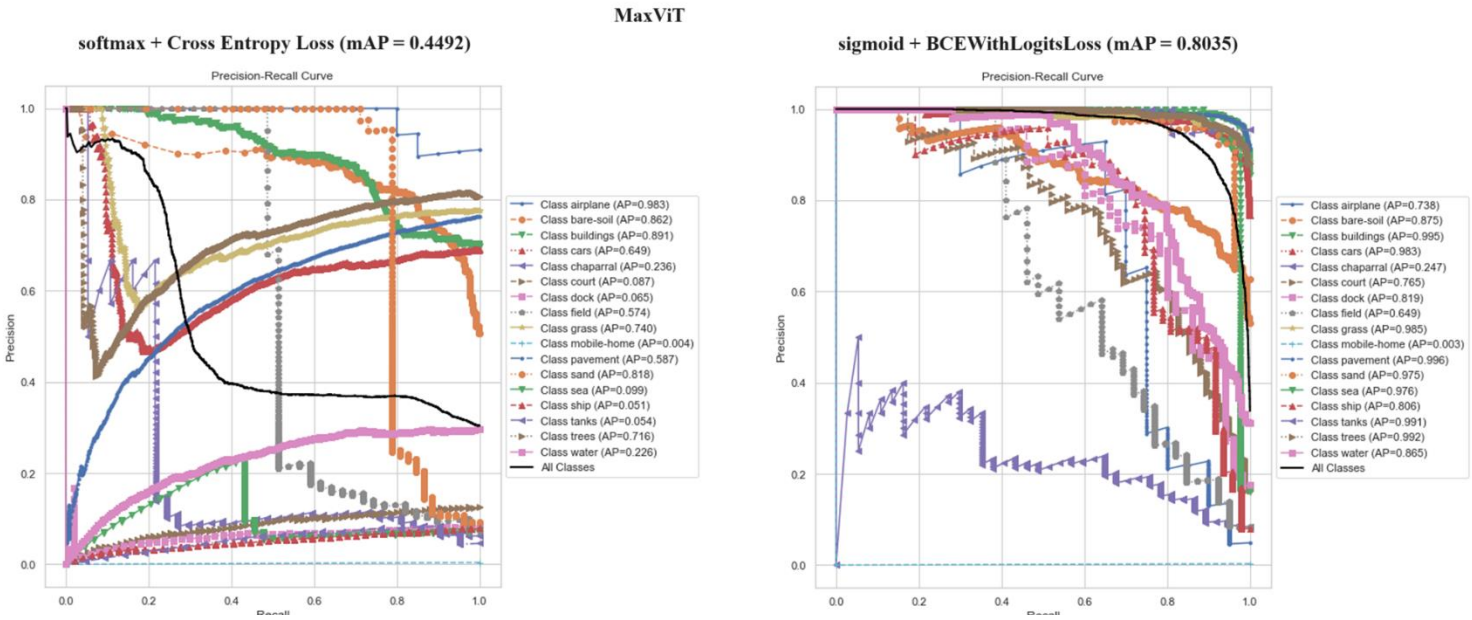


Figure 20. Comparison of using different activation and loss functions for MaxViT

In addition, Ridnik et al. [75] claimed that symmetric loss functions such as focal loss or cross entropy loss is sub-optimal for learning positive samples' features which can be clearly seen on the evaluation metrics of the labels that support results. Activation function for all the models is Sigmoid. Drawing inspiration from this study, ASL is applied as a loss function for the MaxViT model, paired with sigmoid function where ReduceLROnPlaetau is set as the scheduler of the model. γ^+ and γ^- are set as 0 and 4 respectively, since this combination of γ values achieved the most successful result in one of the multi-label image classification studies [82]. Classification report and Precision-Recall curve for all the labels of Multi-label AID dataset can be seen, where γ^+ and γ^- are set as 0 and 4, in Table 10 and Figure 21, respectively.

Table 10. Classification Report of MaxViT, γ^+= 0, γ^-=4, lr=ReduceLROnPlaetau

Class	Precision	Recall	F1-Score
airplane	1	0,55	0,71
bare-soil	0,8	0,84	0,82
buildings	0,95	0,99	0,97
cars	0,92	0,97	0,95

chaparral	0,6	0,08	0,14
court	0,8	0,48	0,6
dock	0,87	0,66	0,75
field	1	0,62	0,76
grass	0,94	0,98	0,96
mobile-home	N/A	N/A	N/A
pavement	0,95	0,99	0,97
sand	0,98	0,88	0,93
sea	1	0,91	0,95
ship	0,71	0,74	0,73
tanks	1	0,9	0,95
trees	0,95	0,96	0,95
water	0,79	0,72	0,76
micro avg	0,92	0,91	0,91
macro avg	0,84	0,72	0,76
weighted avg	0,91	0,91	0,9

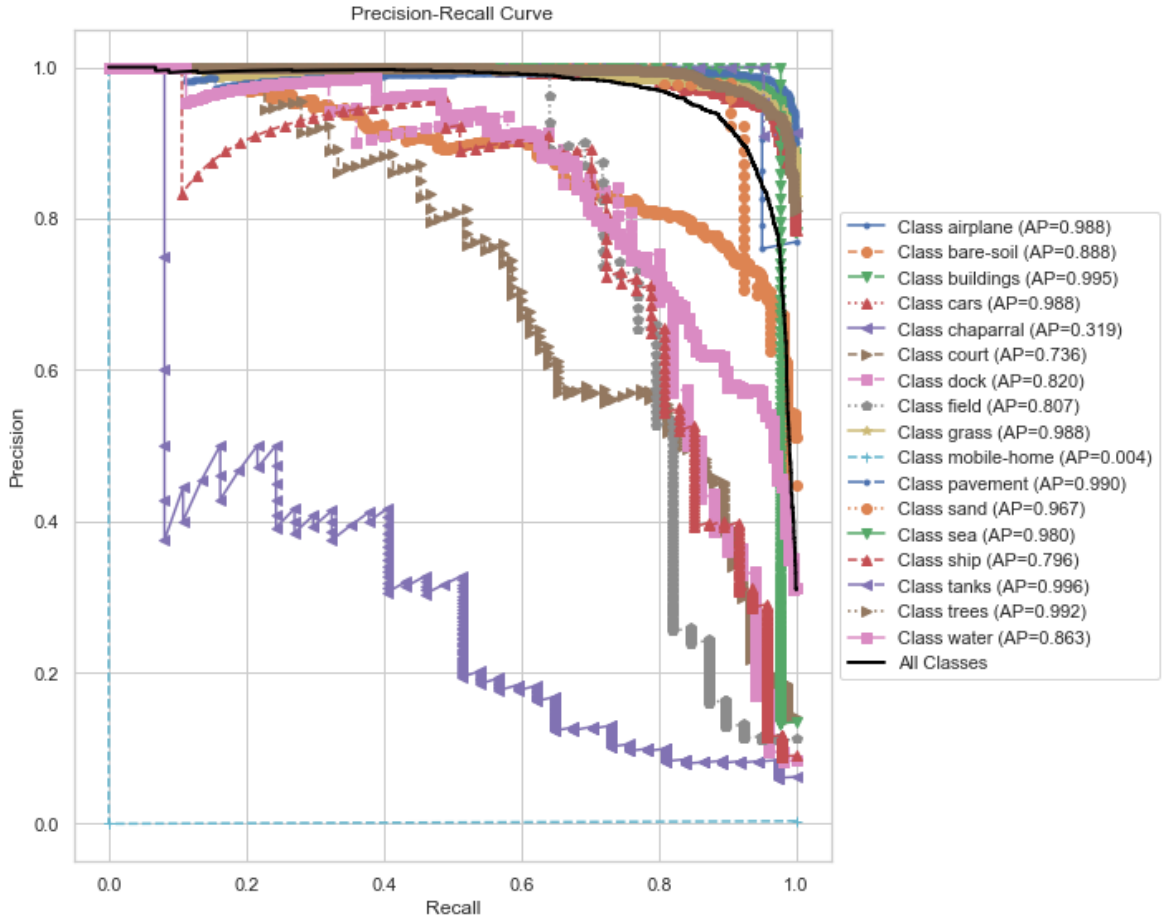


Figure 21. γ^{+}= 0, γ^{-}=4, lr = ReduceLROnPlaetau, mAP = 83.04% for MaxViT

The weighted ASL obviously enhanced the performance amongst widely-used BCEWithLogitsLoss by 2.69%. This increase indicates that γ values, γ^+ and γ^- , have important role in balancing the contribution of positive and negative samples during training. Adjusting both parameters helps giving appropriate importance to the minority class, meanwhile the model does not become biased towards the majority class. Therefore, different values are applied on MaxViT model where mAP is determined as the success criteria (see Table 11).

Table 11. Different γ combinations of mAP (%) on MaxViT for ReduceLROnPlaetau

γ+ / γ-	0	1	2	3	4
2	83.25	**84.12**	83.68	83.60	83.58
4	83.04	82.30	83.20	82.55	82.35

According to the results from Table 11, the most succesful combination of γ^+ and γ^- values are 1 and 2, respectively.

To evaluate the impact of the loss functions, CNN based methods including AlexNet, VGG16, DenseNet-201, Inceptionv3, ConvNeXt and vision transformer based methods including ViT, SwinT and MaxViT are compared. ReduceLROnPlateau is used as a scheduler when the models are compared with two different loss functions; BCEWithLogitsLoss and ASL where γ^+ is set as 1 and γ^- is set as 2 according to the obvious success from Table 11. The results can be seen at Table 12 and Table 13.

Table 12. Asymmetric Loss (γ^+= 1, γ^-=2) , lr = ReduceLROnPlaetau

Object Labels	AlexNet (mAP = 0.7771)			VGG16 (mAP = 0.814)			DenseNet201 (mAP = 0.8441)			Inceptionv3 (mAP = 0.8243)			ConvNeXt (mAP = 0.7881)			ViT (mAP = 0.7938)			SwinT (mAP = 0.8437)			MaxViT (mAP = 0.8412)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
airplane	0,8	0,4	0,53	0,92	0,55	0,69	1	0,65	0,79	0,93	0,65	0,76	1	0,35	0,52	1	0,4	0,57	1	0,65	**0,79**	1	0,6	0,75
bare-soil	0,76	0,78	0,77	0,77	0,81	0,79	0,78	0,82	0,8	0,76	0,81	0,78	0,77	0,8	0,78	0,79	0,81	0,8	0,79	0,85	0,82	0,79	0,85	**0,82**
buildings	0,94	0,96	0,95	0,91	0,97	0,94	0,93	0,99	0,96	0,94	0,97	0,96	0,93	0,98	0,95	0,95	0,98	0,96	0,93	0,98	0,96	0,96	0,98	**0,97**
cars	0,91	0,94	0,92	0,9	0,96	0,93	0,91	0,96	0,93	0,9	0,94	0,92	0,92	0,93	0,93	0,92	0,95	0,93	0,92	0,96	0,94	0,92	0,97	**0,94**
chaparral	0,67	0,05	0,1	1	0,22	**0,36**	0,88	0,19	0,31	0,71	0,14	0,23	1	0,11	0,2	1	0,16	0,28	0,8	0,22	0,34	1	0,14	0,24
court	0,71	0,36	0,48	0,69	0,45	0,55	0,8	0,52	0,63	0,83	0,52	0,64	0,67	0,49	0,57	0,69	0,51	0,58	0,86	0,59	**0,7**	0,84	0,56	0,67
dock	0,76	0,64	0,7	0,81	0,76	0,78	0,84	0,72	0,77	0,81	0,7	0,75	0,81	0,58	0,67	0,92	0,66	0,77	0,82	0,74	**0,78**	0,84	0,72	0,77
field	0,78	0,54	0,64	0,79	0,67	0,72	0,79	0,67	0,72	0,87	0,67	0,75	0,83	0,64	0,72	0,92	0,59	0,72	0,87	0,67	**0,75**	1	0,56	0,72
grass	0,93	0,97	0,95	0,94	0,97	0,96	0,94	0,98	0,96	0,93	0,98	0,96	0,93	0,97	0,95	0,93	0,97	0,95	0,95	0,97	0,96	0,94	0,98	**0,96**
mobile-home	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
pavement	0,93	0,99	0,96	0,95	0,99	0,97	0,95	0,99	0,97	0,95	0,99	0,97	0,95	0,99	0,97	0,94	0,99	0,97	0,95	0,99	0,97	0,97	0,99	**0,98**
sand	0,88	0,88	0,88	0,94	0,87	0,9	0,96	0,88	0,92	0,98	0,9	0,94	0,96	0,88	0,92	0,98	0,9	**0,94**	0,96	0,9	0,93	0,98	0,85	0,91
sea	1	0,93	0,96	0,98	0,95	0,97	0,98	0,91	0,94	0,98	0,95	**0,97**	0,97	0,84	0,9	1	0,86	0,93	1	0,93	0,96	1	0,91	0,95
ship	0,58	0,6	0,59	0,71	0,83	0,76	0,82	0,79	0,8	0,75	0,77	0,76	0,71	0,62	0,66	0,74	0,68	0,71	0,8	0,83	**0,81**	0,82	0,79	0,8
tanks	1	0,95	0,98	0,95	0,95	0,95	1	0,9	0,95	1	0,9	0,95	1	0,81	0,89	1	0,95	0,98	1	0,95	0,98	1	0,95	**0,98**
trees	0,93	0,96	0,94	0,93	0,95	0,94	0,92	0,96	0,94	0,93	0,95	0,94	0,94	0,97	0,95	0,92	0,97	0,94	0,94	0,96	0,95	0,94	0,97	**0,96**
water	0,68	0,66	0,67	0,74	0,69	0,72	0,79	0,7	0,74	0,78	0,7	0,74	0,72	0,67	0,69	0,71	0,66	0,68	0,79	0,74	0,76	0,79	0,76	**0,78**
micro avg	0,89	0,88	0,88	0,89	0,9	0,9	0,9	0,91	0,91	0,9	0,9	0,9	0,9	0,89	0,89	0,9	0,89	0,9	0,91	0,91	0,91	0,92	0,91	0,92
macro avg	0,78	0,68	0,71	0,82	0,74	0,76	0,84	0,74	0,77	0,83	0,74	0,77	0,83	0,68	0,72	0,85	0,71	0,75	0,85	0,76	0,79	0,87	0,74	0,78
weighted avg	0,88	0,88	0,88	0,89	0,9	0,89	0,9	0,91	0,9	0,9	0,9	0,9	0,89	0,89	0,89	0,9	0,89	0,89	0,91	0,91	0,91	0,92	0,91	0,91

Table 13. BCEWithLogitsLoss, lr = ReduceLROnPlaetau

Object Labels	AlexNet (mAP = 0.7801)			VGG16 (mAP = 0.8063)			DenseNet201 (mAP = 0.8272)			Inceptionv3 (mAP = 0.8233)			ConvNeXt (mAP = 0.7745)			ViT (mAP = 0.7749)			SwinT (mAP = 0.8422)			MaxViT (mAP = 0.8302)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
airplane	1	0,35	0,52	0,91	0,5	0,65	1	0,6	0,75	1	0,7	**0,82**	1	0,15	0,26	1	0,15	0,26	1	0,65	0,79	1	0,5	0,67
bare-soil	0,77	0,74	0,75	0,75	0,78	0,76	0,79	0,82	0,8	0,78	0,77	0,78	0,77	0,8	0,78	0,78	0,81	0,79	0,83	0,81	**0,82**	0,8	0,8	0,8
buildings	0,95	0,95	0,95	0,93	0,97	0,95	0,94	0,97	0,96	0,93	0,97	0,95	0,94	0,97	0,95	0,98	0,96	**0,97**	0,95	0,97	0,96	0,96	0,96	0,96
cars	0,92	0,93	0,93	0,93	0,96	0,94	0,91	0,96	0,93	0,9	0,95	0,93	0,92	0,93	0,93	0,92	0,93	0,93	0,93	0,95	0,94	0,94	0,95	**0,94**
chaparral	0,67	0,05	0,1	0,6	0,08	0,14	0,57	0,11	0,18	0,86	0,16	0,27	0,5	0,08	0,14	0,57	0,11	0,18	0,89	0,22	**0,35**	0,5	0,03	0,05
court	0,62	0,32	0,42	0,81	0,4	0,54	0,83	0,57	0,68	0,75	0,48	0,59	0,76	0,39	0,51	0,78	0,48	0,6	0,8	0,49	0,61	0,87	0,53	**0,66**
dock	0,79	0,6	0,68	0,83	0,7	0,76	0,86	0,62	0,72	0,81	0,68	0,74	0,83	0,6	0,7	0,94	0,58	0,72	0,85	0,68	0,76	0,85	0,7	**0,77**
field	0,89	0,62	0,73	0,86	0,62	0,72	0,89	0,64	0,75	0,96	0,67	**0,79**	0,86	0,62	0,72	0,84	0,54	0,66	0,85	0,59	0,7	1	0,51	0,68
grass	0,94	0,96	0,95	0,94	0,96	0,95	0,95	0,97	0,96	0,94	0,97	0,95	0,94	0,98	0,96	0,94	0,95	0,95	0,95	0,95	0,95	0,96	0,95	**0,96**
mobile-home	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
pavement	0,95	0,99	0,97	0,95	0,99	0,97	0,96	0,99	0,98	0,96	0,99	0,97	0,95	0,99	0,97	0,96	0,98	0,97	0,96	0,99	**0,98**	0,96	0,99	0,97
sand	0,91	0,83	0,87	0,92	0,87	0,87	0,96	0,88	0,92	0,92	0,87	0,89	0,98	0,85	0,91	0,96	0,88	0,92	0,96	0,88	**0,92**	0,95	0,81	0,88
sea	0,97	0,86	0,92	0,95	0,86	0,9	1	0,91	0,95	0,98	0,95	**0,97**	0,97	0,84	0,9	0,97	0,89	0,93	1	0,93	0,96	1	0,93	0,96
ship	0,68	0,6	0,64	0,73	0,79	0,76	0,81	0,74	0,78	0,74	0,72	0,73	0,71	0,64	0,67	0,76	0,72	0,74	0,86	0,77	**0,81**	0,76	0,72	0,74
tanks	0,95	0,9	0,93	1	0,95	0,98	1	0,95	0,98	1	0,9	0,95	1	0,81	0,89	1	0,95	0,98	0,95	0,95	0,95	1	0,95	**0,98**
trees	0,94	0,95	0,94	0,93	0,95	0,94	0,94	0,96	0,95	0,93	0,95	0,94	0,94	0,95	0,94	0,93	0,95	0,94	0,94	0,96	0,95	0,95	0,95	**0,95**
water	0,7	0,64	0,67	0,75	0,71	0,73	0,84	0,7	0,76	0,83	0,7	0,76	0,71	0,68	0,69	0,76	0,62	0,68	0,82	0,76	**0,79**	0,82	0,69	0,75
micro avg	0,9	0,87	0,88	0,9	0,89	0,89	0,92	0,9	0,91	0,91	0,89	0,9	0,9	0,88	0,89	0,91	0,88	0,9	0,93	0,9	0,91	0,93	0,89	**0,91**
macro avg	0,8	0,66	0,7	0,81	0,71	0,74	0,84	0,73	0,77	0,84	0,73	0,77	0,81	0,66	0,7	0,83	0,68	0,72	0,86	0,74	**0,78**	0,84	0,7	0,75
weighted avg	0,89	0,87	0,87	0,89	0,89	0,89	0,91	0,9	0,9	0,9	0,89	0,89	0,89	0,88	0,88	0,91	0,88	0,89	0,92	0,9	**0,91**	0,92	0,89	0,9

Due to the comparison of Table 12 and Table 13, BCEWithLogitsLoss is slightly more successful than ASL, but only with AlexNet. The evaluation metrics show that the models that use ASL as a loss function classify the images better than the models that use BCEWithLogitsLoss.

In deep learning applications, a single hyperparameter does not always play the main role. A scheduler is another key factor that significantly contributes to the model's success in classification tasks. OneCycleLR adjusts the learning rate in a cyclical pattern during training, where it rises to a peak value and then decreases, as illustrated in Figure 18. This approach can enhance overall convergence. The effectiveness of OneCycleLR is evaluated on the same models using the two previously mentioned loss functions. The outcomes are presented in Table 14 and Table 15.

Table 14. Asymmetric Loss (γ^+= 1, γ^-=2), lr = OneCycleLR

Object Labels	AlexNet (mAP = 0.727)			VGG16 (mAP = 0.7411)			DenseNet201 (mAP = 0.7965)			Inceptionv3 (mAP = 0.8183)			ConvNeXt (mAP = 0.7938)			ViT (mAP = 0.795)			SwinT (mAP = 0.7994)			MaxViT (mAP = 0.8498)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
airplane	0,69	0,45	0,55	0,5	0,25	0,33	0,68	0,65	0,67	1	0,6	0,75	1	0,3	0,46	1	0,15	0,26	1	0,55	0,71	1	0,75	**0,86**
bare-soil	0,75	0,63	0,68	0,73	0,69	0,71	0,79	0,77	0,78	0,75	0,85	0,8	0,78	0,78	0,78	0,79	0,83	**0,81**	0,81	0,65	0,72	0,79	0,82	0,8
buildings	0,91	0,98	0,94	0,9	0,98	0,94	0,87	0,99	0,92	0,91	0,99	0,94	0,94	0,98	0,96	0,92	0,99	0,95	0,94	0,97	0,95	0,94	0,98	**0,96**
cars	0,89	0,94	0,92	0,9	0,94	0,92	0,88	0,97	0,92	0,88	0,96	0,92	0,91	0,94	0,92	0,9	0,97	0,93	0,89	0,97	0,93	0,92	0,96	**0,94**
chaparral	0,71	0,14	0,23	0,8	0,11	0,19	1	0,14	0,24	0,88	0,19	0,31	0,67	0,05	0,1	1	0,16	0,28	0,59	0,27	**0,37**	0,86	0,16	0,27
court	0,66	0,33	0,44	0,83	0,45	0,59	0,78	0,41	0,54	0,47	0,79	0,59	0,65	0,43	0,52	0,71	0,47	0,56	0,76	0,6	0,67	0,85	0,59	**0,69**
dock	0,63	0,52	0,57	0,71	0,48	0,57	0,58	0,74	0,65	0,66	0,82	0,73	0,76	0,58	0,66	0,86	0,72	0,78	0,81	0,78	**0,8**	0,8	0,7	0,74
field	0,71	0,38	0,5	0,84	0,54	0,66	0,82	0,46	0,59	0,79	0,59	0,68	0,89	0,62	0,73	0,92	0,62	0,74	0,81	0,64	0,71	0,9	0,67	**0,76**
grass	0,92	0,95	0,94	0,94	0,93	0,94	0,93	0,96	0,94	0,94	0,96	0,95	0,93	0,97	0,95	0,92	0,97	0,95	0,94	0,95	0,94	0,94	0,98	**0,96**
mobile-home	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
pavement	0,93	0,97	0,95	0,94	0,99	0,97	0,9	1	0,95	0,93	1	0,96	0,95	0,99	0,97	0,94	1	0,97	0,94	0,99	0,97	0,97	0,99	**0,98**
sand	0,83	0,92	0,87	0,92	0,44	0,6	0,92	0,88	0,9	0,96	0,88	**0,92**	0,96	0,85	0,9	0,98	0,85	0,91	0,98	0,81	0,88	0,94	0,87	0,9
sea	0,94	0,75	0,84	0,88	0,86	0,87	1	0,86	0,93	0,87	0,93	0,9	1	0,84	0,91	1	0,84	0,91	1	0,91	0,95	1	0,95	**0,98**
ship	0,51	0,47	0,49	0,76	0,66	0,7	0,71	0,72	0,72	0,53	0,89	0,66	0,69	0,62	0,65	0,7	0,7	0,7	0,65	0,77	0,71	0,82	0,87	**0,85**
tanks	0,82	0,86	0,84	1	0,86	0,92	1	0,9	0,95	1	0,86	0,92	1	0,9	0,95	1	0,95	0,98	0,89	0,81	0,85	1	0,95	**0,98**
trees	0,93	0,98	0,95	0,94	0,93	0,94	0,92	0,96	0,94	0,94	0,95	0,95	0,93	0,97	0,95	0,94	0,97	0,95	0,95	0,94	0,94	0,95	0,97	**0,96**
water	0,66	0,62	0,64	0,78	0,64	0,7	0,83	0,66	0,74	0,84	0,64	0,73	0,64	0,7	0,66	0,7	0,71	0,71	0,73	0,69	0,71	0,81	0,74	**0,78**
micro avg	0,87	0,86	0,86	0,89	0,85	0,87	0,88	0,89	0,88	0,87	0,91	0,89	0,89	0,89	0,89	0,89	0,9	0,9	0,9	0,88	0,89	0,92	0,91	**0,91**
macro avg	0,73	0,64	0,67	0,79	0,63	0,68	0,8	0,71	0,73	0,79	0,76	0,75	0,81	0,68	0,71	0,84	0,7	0,73	0,81	0,72	0,75	0,85	0,76	**0,79**
weighted avg	0,86	0,86	0,85	0,88	0,85	0,86	0,88	0,89	0,88	0,88	0,91	0,89	0,88	0,89	0,88	0,89	0,9	0,89	0,89	0,88	0,89	0,91	0,91	**0,91**

Table 15. BCEWithLogitsLoss, lr = OneCycleLR

Object Labels	AlexNet (mAP = 0.7379)			VGG16 (mAP = 0.7616)			DenseNet201 (mAP = 0.8009)			Inceptionv3 (mAP = 0.8063)			ConvNeXt (mAP = 0.7270)			ViT (mAP = 0.7966)			SwinT (mAP = 0.7888)			MaxViT (mAP = 0.8035)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
airplane	0,48	0,65	0,55	1	0,1	0,18	0,92	0,55	0,69	1	0,6	0,75	0	0	0	1	0,25	0,4	1	0,65	**0,79**	0,92	0,55	0,69
bare-soil	0,78	0,74	0,76	0,8	0,55	0,65	0,76	0,84	0,8	0,8	0,73	0,76	0,75	0,81	0,78	0,79	0,85	**0,81**	0,79	0,81	0,8	0,78	0,79	0,78
buildings	0,95	0,94	0,94	0,93	0,96	0,94	0,95	0,96	0,95	0,93	0,98	0,95	0,98	0,94	0,96	0,95	0,97	0,96	0,93	0,99	0,95	0,97	0,97	**0,97**
cars	0,94	0,88	0,91	0,91	0,93	0,92	0,93	0,93	0,93	0,9	0,97	0,94	0,93	0,92	0,92	0,95	0,93	**0,94**	0,92	0,91	0,91	0,95	0,91	0,93
chaparral	0,6	0,08	0,14	0,67	0,05	0,1	0,43	0,16	0,24	0,75	0,08	0,15	0	0	0	0,75	0,16	0,27	0,86	0,16	0,27	0,35	0,3	**0,32**
court	0,7	0,35	0,46	0,84	0,35	0,49	0,74	0,53	0,62	0,88	0,39	0,54	0,78	0,33	0,47	0,83	0,4	0,54	0,83	0,33	0,48	0,54	0,83	**0,66**
dock	0,56	0,74	0,64	0,9	0,52	0,66	0,67	0,76	0,71	0,74	0,74	**0,74**	0,61	0,66	0,63	0,93	0,5	0,65	0,88	0,44	0,59	0,9	0,52	0,66
field	0,54	0,69	0,61	0,95	0,51	0,67	0,74	0,64	0,68	0,77	0,62	0,69	1	0,28	0,44	0,92	0,62	**0,74**	0,85	0,44	0,58	0,89	0,41	0,56
grass	0,93	0,95	0,94	0,93	0,95	0,94	0,93	0,97	0,95	0,95	0,94	0,94	0,95	0,93	0,94	0,94	0,97	0,95	0,93	0,96	0,95	0,94	0,96	**0,95**
mobile-home	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
pavement	0,93	0,98	0,96	0,96	0,98	0,97	0,95	0,98	0,97	0,95	0,99	0,97	0,95	0,98	0,96	0,97	0,99	**0,98**	0,94	0,99	0,96	0,97	0,97	0,97
sand	0,9	0,87	0,88	0,87	0,87	0,87	0,85	0,88	0,87	0,85	0,87	0,86	0,79	0,81	0,8	0,98	0,85	**0,91**	0,98	0,77	0,86	0,98	0,77	0,86
sea	0,93	0,86	0,89	1	0,7	0,83	0,98	0,91	0,94	0,95	0,93	**0,94**	0,82	0,91	0,86	1	0,75	0,86	1	0,59	0,74	0,95	0,91	0,93
ship	0,56	0,74	0,64	0,82	0,68	**0,74**	0,68	0,77	0,72	0,62	0,85	0,71	0,56	0,66	0,61	0,85	0,6	0,7	0,81	0,47	0,59	0,9	0,57	0,7
tanks	0,95	0,95	0,95	0,89	0,76	0,82	1	0,9	0,95	0,95	0,86	0,9	1	0,43	0,6	1	0,95	**0,98**	1	0,76	0,86	0,91	1	0,95
trees	0,94	0,94	0,94	0,93	0,95	0,94	0,94	0,95	0,95	0,96	0,93	0,94	0,95	0,94	0,94	0,95	0,97	**0,96**	0,91	0,97	0,94	0,96	0,94	0,95
water	0,64	0,67	0,66	0,74	0,66	0,7	0,75	0,76	**0,76**	0,9	0,62	0,73	0,59	0,47	0,53	0,66	0,7	0,68	0,87	0,62	0,72	0,92	0,61	0,73
micro avg	0,88	0,87	0,87	0,91	0,85	0,88	0,89	0,9	0,9	0,91	0,88	0,89	0,89	0,85	0,87	0,91	0,89	0,9	0,91	0,87	0,89	0,91	0,88	**0,9**
macro avg	0,73	0,71	0,7	0,83	0,62	0,67	0,78	0,74	**0,75**	0,82	0,71	0,74	0,69	0,59	0,61	0,85	0,67	0,72	0,85	0,64	0,71	0,81	0,71	0,74
weighted avg	0,88	0,87	0,87	0,9	0,85	0,86	0,89	0,9	0,89	0,91	0,88	0,88	0,87	0,85	0,85	0,91	0,89	0,89	0,91	0,87	0,88	0,92	0,88	**0,89**

The comparison of Table 14 and Table 15 shows that BCEWithLogitsLoss is slightly better than ASL in CNN models except Inceptionv3 and ConvNeXt. They align with binary classification objectives and are robust in handling data imbalances by providing stable probability outputs. These factors together lead to improved precision and recall, and results in higher mAP scores. Conversely, ASL performs better than BCEWithLogitsLoss for the vision transformer models, except a small difference in ViT. ASL handles the class imbalances more effectively by weighting positive and negative examples differently that deals with imbalanced datasets, such as Multi-label AID, where one class is underrepresented. For example, airplane and chaparral have less instances when compared to the other labels, 20 and 37 respectively (see Table 2). The success of combining ASL and OneCycleLR on vision transformer models is undeniable, as they achieved the highest F1-Score in 16 out of a total of 17 classes where F1-Score is more comprehensive and critical, considering recall and precision metrics. The comparison of MaxViT model with two loss functions, it is clear that ASL weights the positive and negative samples much better than BCEWithLogitsLoss. Handling imbalanced data in a more successful way also lead MaxViT model to increase mAP value by 4.63%. As such, the other models achieved much better results on these two specific labels when ASL is used as a loss function, regardless of overall mAP success or the scheduler type.

Just like Table 11, different γ values are evaluated on the most successful model, MaxViT, where the scheduler is OneCycleLR. The parameters that were chosen in Table 11,

using ReduceLROnPlateau as a scheduler, are still the best combination where OneCycleLR is assigned as a scheduler. The different mAP values of ASL are given in Table 16.

Table 16. Impact of different γ combinations of mAP (%) on MaxViT for OneCycleLR

γ- \ γ+	0	1	2	3	4
2	84.2	**84.98**	83.82	83.46	84.41
4	83.58	84.03	83.83	82.95	83.92

Table 16 shows that different combinations of γ have a noticeable impact on the various approaches..

MaxViT is an architecture that combines the convolutional operations and attention mechanisms. Apart from this initially created network, a fusion model is created by selecting the most successful CNN-based network and vision transformer based network. By comparing the results from Table 12, Table 13, Table 14 and Table 15; SwinT (84.37%) and DenseNet-201 (84.41%) are combined for the fusion model. As can be seen in Table 12, each model individually demonstrates superior performance compared to MaxViT with mAP of 84.12%.

Pre-trained SwinT and DenseNet-201 models are utilized, and feature extractors are set up to capture features from images in the Multi-label AID dataset. The feature extractor retrieves parts of DenseNet-201, including all convolutional layers up to the classification layer. A feature map of size (1920, 7, 7) is extracted, where 1920 represents the number of channels, and 7 x 7 represents the spatial dimensions. Similarly, another extractor captures features from SwinT, which consists of all convolutional layers except for the classification layer, as with the DenseNet-201 extractor. The features from SwinT have 768 channels, corresponding to the final transformer's output. These features do not retain the spatial dimension in SwinT, maintaining spatial relationships through local and shifted windows. As the model processes deeper layers, the spatial resolution reduces, and the feature representation becomes more abstract and higher-level. In the subsequent step of the fusion model, feature maps from both models are converted into 1D vectors and combined. The combined features are then fed into a dense layer with a dropout rate of 0.2 for classification purposes. The classification outcomes are presented in Table 17 and Figure 22.

Table 17. Classification report of the fusion model

Class	Precision	Recall	F1-Score
airplane	1	0,5	0,67
bare-soil	0,78	0,76	0,77
buildings	0,93	0,97	0,95
cars	0,91	0,97	0,94
chaparral	0,67	0,11	0,19
court	0,7	0,49	0,58
dock	0,84	0,72	0,77
field	0,96	0,62	0,75
grass	0,95	0,97	0,96
mobile-home	N/A	N/A	N/A
pavement	0,94	0,99	0,97
sand	0,92	0,85	0,88
sea	0,97	0,89	0,93
ship	0,79	0,81	0,8
tanks	0,95	0,86	0,9
trees	0,93	0,97	0,95
water	0,86	0,73	0,79
micro avg	0,91	0,9	0,9
macro avg	0,83	0,72	0,75
weighted avg	0,9	0,9	0,9

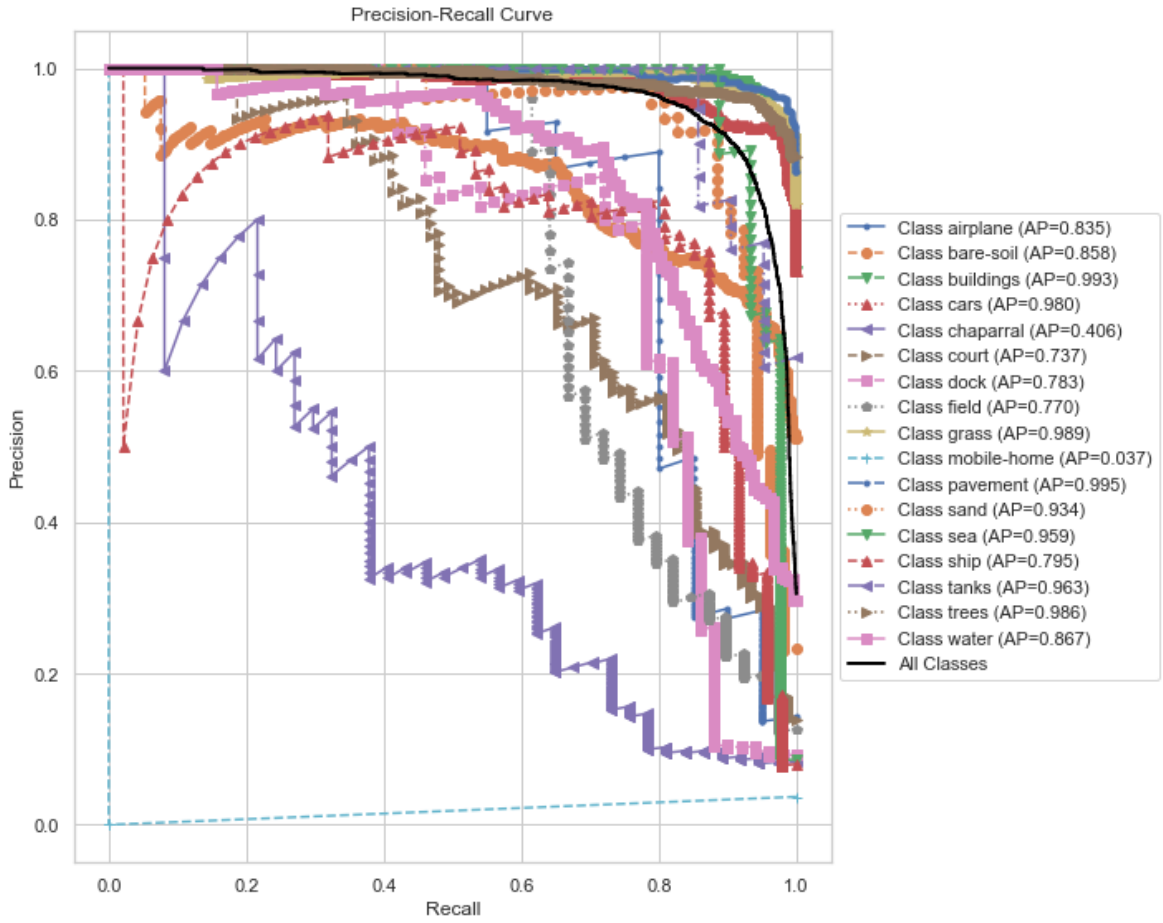


Figure 22. Precision-recall curve of the fusion model

The fusion model achieved mAP of 81.68%, which is way below of the success from MaxViT (84.98%). The features that are extracted from both models might not be compatible, combining them together might lead to suboptimal performance. MaxViT performs both grid and global attention mechanisms (see Figure 16) that allows the model to capture multi-scale features more effectively than a simple fusion of CNN and transformer models. As far as is known, mAP of 84.98% has been achieved for the first time on Multi-label AID dataset. A comparison of performance metrics between earlier research and MaxViT is given in Table 18.

Table 18. A comparison of performance metrics utilizing the Multi-label AID dataset

Method	Score (CF1/CP/CR)	mAP (%)
AL-RN-ResNet50 [66]	88.72 / 91.00 / 88.95	-
MLRSSC-CNN-GNN [18]	88.64 / 89.83 / 90.20	-
ResNet50-SR-Net [67]	89.97 / 89.42 / 90.52	-
S-MAT-ResNet50 [68]	90.90 / 92.17 / 89.69	-

LD-GCN [69]	90.93 / **92.81** / 89.06	83.49
MaxViT	**91** / 92 / **91**	**84.98**

The success of MaxViT can be explained by its architecture since both convolutional operations and multi-head self-attention are leveraged. This dual capability enables both local and global features to be captured effectively and particularly beneficial for multi-label classification tasks where different labels might correspond to distinct features at various scales and regions within an image. Spatial relationships and local patterns within the images of Multi-label AID are captured by MaxViT which facilitates hierarchical feature extraction. The features are extracted at different levels of abstraction, from edges and textures in the initial layers to more complex shapes and objects in the deeper layers. Extracted initial feature maps, which contains localized and spatially-aware representations of the image, serve as input for the subsequent transformer blocks. These input feature maps processed by transformer blocks using self-attention and FFN. Global features are captured by the self-attention mechanism in each MaxViT block, with the importance of each feature being computed relative to all others in the feature map. FFN processes the combined local and global features, adds non-linearity. As a result, the extraction of local features through convolutional layers and global features via the multi-head self-attention mechanism creates a powerful and flexible architecture, enhancing the model's capability to identify and distinguish multiple labels within a single image. In addition, the integration of ASL and OneCycleLR not only improved mAP value, but also showed robustness in handling the complexities of multi-label image classification. Precision-Recall curve for all the labels can be seen in Figure 23, using MaxViT where γ^+ and γ^- are set as 1 and 2, respectively.

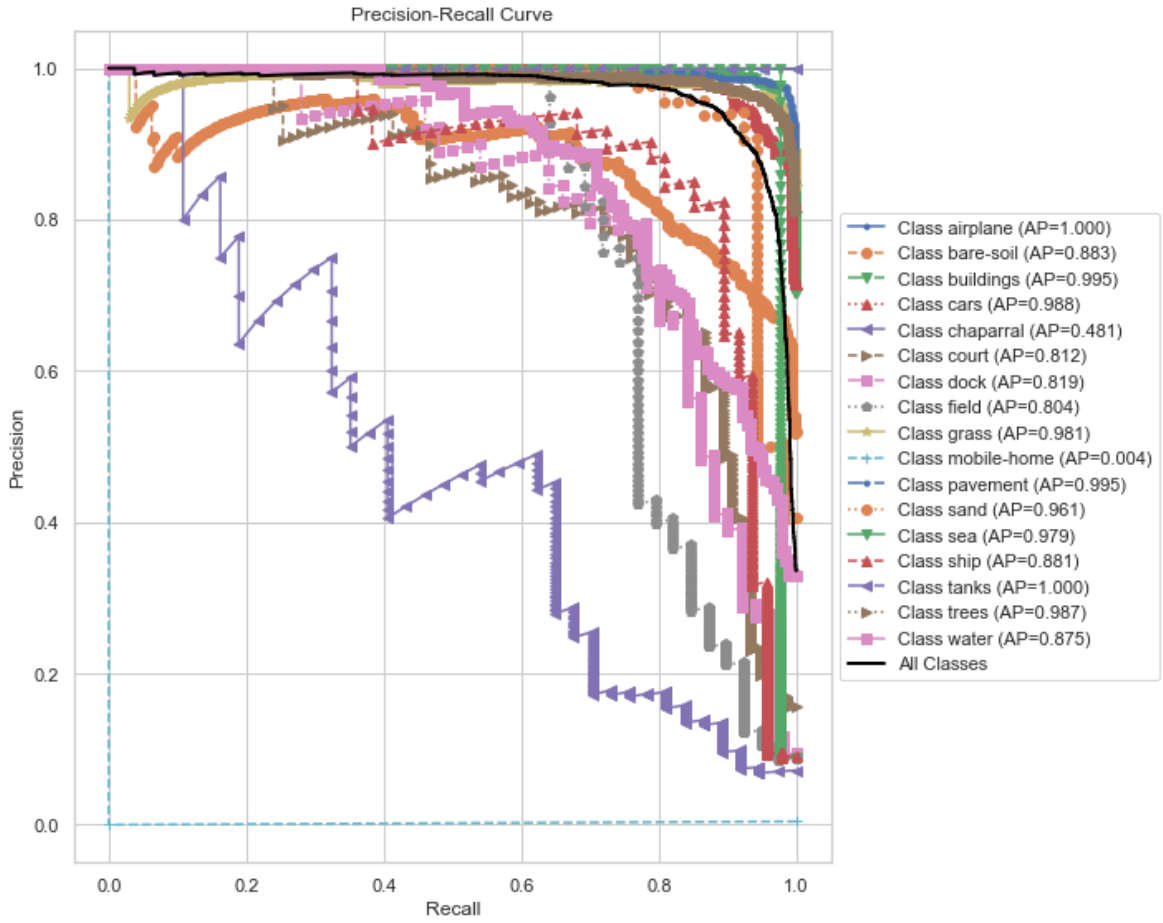


Figure 23. γ^{+}= 1, γ^{-}=2, lr = OneCycleLR, mAP = 84.98% for MaxViT

6. CONCLUSION

The multi-label AID dataset has some shortcomings. Some classes have significantly fewer classes. As can be seen in Table 2, chaparral has 37 instances, airplane has 20 instances and mobile-home has only 1 instance. This leads to lower precision and recall, therefore the average precision calculation is directly affected. Especially for the mobile-home, not even a single model is able to perform any classification due to the lack of instance. The maximum mAP value that all the labels can achieve without further improvement is 94.12%, due to the single instance for this specific label. There are also some classes like court, dock or field that the model shows moderate performance due to the instance quantity. Even if the precision is relatively high, lower recall results for the model to miss some true positives. Classes with lower recall value drag down the average precision even if the other classes perform well. Coupling of ASL which is designed to handle class imbalance effectively, and OneCycleLR that provides dynamic learning rate adjustments improved overall model accuracy. Among the different configurations tested, the combination of ASL and OneCycleLR scheduler performed the most successful outcomes on MaxViT-T model. The findings demonstrate that the choice of loss function and learning rate scheduler in an appropriate network model impacts the performance of the model significantly.

In the next studies, classification performance can be increased by focusing on achieving high classification rates on small instances and finding the best combination of the loss function, scheduler and window-based vision transformer. Due to the limited computational resources, the MaxViT-L algorithm could not be implemented in this thesis. In future applications, this algorithm can be adopted for better classification results. In addition, the features that are extracted from both models (SwinT and DenseNet-201) might not be compatible, combining them together might lead to suboptimal performance. Further fusion models might require additional fine-tuning to work harmoniously together.

REFERENCES

[1] Y. Zik and G. Hon, “History of science and science combined: solving a historical problem in optics—the case of Galileo and his telescope,” *Arch Hist Exact Sci*, vol. 71, no. 4, pp. 337–344, Jul. 2017, doi: 10.1007/s00407-016-0187-y.

[2] “Thomas J. J. See,” *Phys Today*, vol. 15, no. 8, p. 80, Aug. 1962, doi: 10.1063/1.3058355.

[3] “An AeriAl Pax americana 1 enrique rAmirez.”

[4] S. Christe, B. Zeiger, R. Pfaff, and M. Garcia, “Introduction to the Special Issue on Sounding Rockets and Instrumentation,” Feb. 2016

[5] “The Space Shuttle Decision,” 1999.

[6] D. K. Sinha, P. P. Chakraborty, A. Rahman, A. Kumar Saha, and A. R. Siddiqui, “Remote Sensing and GIS Module: Basic Physics of Remote Sensing.”

[7] M. Citation: Wójtowicz, A. Wójtowicz, and J. Piekarczyk, “Application of remote sensing methods in agriculture,” 2016.

[8] G. Koukiou, “SAR Features and Techniques for Urban Planning—A Review,” Jun. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/rs16111923.

[9] J. S. Estrada, A. Fuentes, P. Reszka, and F. Auat Cheein, “Machine learning assisted remote forestry health assessment: a comprehensive state of the art review,” 2023, *Frontiers Media S.A.* doi: 10.3389/fpls.2023.1139232.

[10] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, “Google Earth Engine: Planetary-scale geospatial analysis for everyone,” *Remote Sens Environ*, vol. 202, pp. 18–27, Dec. 2017, doi: 10.1016/j.rse.2017.06.031.

[11] Y. Li, Y. Zhang, X. Huang, and A. L. Yuille, “Deep networks under scene-level supervision for multi-class geospatial object detection from remote sensing images,”

ISPRS Journal of Photogrammetry and Remote Sensing, vol. 146, pp. 182–196, Dec. 2018, doi: 10.1016/j.isprsjprs.2018.09.014.

[12] Y. Li, J. Ma, and Y. Zhang, "Image retrieval from remote sensing big data: A survey," *Information Fusion*, vol. 67, pp. 94–115, Mar. 2021, doi: 10.1016/j.inffus.2020.10.008.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," Institute of Electrical and Electronics Engineers (IEEE), Mar. 2010, pp. 248–255. doi: 10.1109/cvpr.2009.5206848.

[14] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories."

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks."

[16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014

[17] C. Szegedy *et al.*, "Going Deeper with Convolutions," Sep. 2014

[18] Y. Li, R. Chen, Y. Zhang, M. Zhang, and L. Chen, "Multi-label remote sensing image scene classification by combining a convolutional neural network and a graph neural network," *Remote Sens (Basel)*, vol. 12, no. 23, pp. 1–17, Dec. 2020, doi: 10.3390/rs12234003.

[19] J. Bogatinovski, L. Todorovski, S. Džeroski, and D. Kocev, "Comprehensive comparative study of multi-label classification methods," *Expert Syst Appl*, vol. 203, Oct. 2022, doi: 10.1016/j.eswa.2022.117215.

[20] S. Narwane and S. Sawarkar, "Machine Learning and Class Imbalance: A Literature Survey," *Industrial Engineering Journal*, vol. 12, no. 10, Oct. 2019, doi: 10.26488/iej.12.10.1202.

[21] W. Bi and J. T. Kwok, "Multilabel Classification with Label Correlations and Missing Labels."

[22] W. Siblini, P. Kuntz, and F. Meyer, “A review on dimensionality reduction for multi-label classification,” *IEEE Trans Knowl Data Eng*, 2019, doi: 10.1109/TKDE.2019.2940014ï.

[23] E. Ben-Baruch *et al.*, “Multi-label Classification with Partial Annotations using Class-aware Selective Loss.”

[24] C. Guo *et al.*, “Breaking the Glass Ceiling for Embedding-Based Classifiers for Large Output Spaces.”

[25] Y. J. Lin and C. J. Lin, “On the Thresholding Strategy for Infrequent Labels in Multi-label Classification,” in *International Conference on Information and Knowledge Management, Proceedings*, Association for Computing Machinery, Oct. 2023, pp. 1441–1450. doi: 10.1145/3583780.3614996.

[26] T. A. Odetola, O. Oderhohwo, and S. R. Hasan, “A Scalable Multilabel Classification to Deploy Deep Learning Architectures For Edge Devices,” Nov. 2019

[27] L. Mou and X. X. Zhu, “Vehicle Instance Segmentation from Aerial Image and Video Using a Multi-Task Learning Residual Fully Convolutional Network,” May 2018, doi: 10.1109/TGRS.2018.2841808.

[28] S. Lucchesi, M. Giardino, and L. Perotti, “Applications of high-resolution images and DTMs for detailed geomorphological analysis of mountain and plain areas of NW Italy,” *Eur J Remote Sens*, vol. 46, no. 1, pp. 216–233, 2013, doi: 10.5721/EuJRS20134612.

[29] N. Audebert, B. Le Saux, and S. Lefèvre, “Fusion of Heterogeneous Data in Convolutional Networks for Urban Semantic Labeling (Invited Paper),” Jan. 2017

[30] P. Helber, B. Bischke, A. Dengel, and D. Borth, “EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification,” Aug.

[31] N. Ali and B. Zafar, “RSSCN7 Image dataset,” Sep. 2018, *figshare*. doi: 10.6084/m9.figshare.7006946.v1.

[32] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270–279.

[33] G. S. Xia *et al.*, “AID: A benchmark data set for performance evaluation of aerial scene classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.

[34] G. Cheng, J. Han, S. Member, and X. Lu, “Remote Sensing Image Scene Classification: Benchmark and State of the Art.”

[35] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” Sep. 2014

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.”

[37] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with Noisy Student improves ImageNet classification,” Nov. 2019

[38] H. Zhang *et al.*, “ResNeSt: Split-Attention Networks,” Apr. 2020

[39] B. Wu *et al.*, “Visual Transformers: Token-based Image Representation and Processing for Computer Vision,” Jun. 2020

[40] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Oct. 2020

[41] A. Rangel, J. Terven, D. M. Cordova-Esparza, and E. A. Chavez-Urbiola, “Land Cover Image Classification,” Jan. 2024

[42] Z. Tu *et al.*, “MaxViT: Multi-Axis Vision Transformer,” Apr. 2022

[43] L. Y. Niu, Y. Wei, and Y. Liu, “Event-driven spiking neural network based on membrane potential modulation for remote sensing image classification,” *Eng Appl Artif Intell*, vol. 123, Aug. 2023, doi: 10.1016/j.engappai.2023.106322.

[44] H. Yue, L. Qing, Z. Zhang, Z. Wang, L. Guo, and Y. Peng, “MSE-Net: A novel master–slave encoding network for remote sensing scene classification,” *Eng Appl Artif Intell*, vol. 132, Jun. 2024, doi: 10.1016/j.engappai.2024.107909.

[45] Z. Liu *et al.*, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” Mar. 2021

[46] M. Kaselimi, A. Voulodimos, I. Daskalopoulos, N. Doulamis, and A. Doulamis, “A Vision Transformer Model for Convolution-Free Multilabel Classification of Satellite Imagery in Deforestation Monitoring,” *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 7, pp. 3299–3307, Jul. 2023, doi: 10.1109/TNNLS.2022.3144791.

[47] F. Wang, J. Ji, and Y. Wang, “DSViT: Dynamically Scalable Vision Transformer for Remote Sensing Image Segmentation and Classification,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 16, pp. 5441–5452, 2023, doi: 10.1109/JSTARS.2023.3285259.

[48] P. Lv, W. Wu, Y. Zhong, F. Du, and L. Zhang, “SCViT: A Spatial-Channel Feature Preserving Vision Transformer for Remote Sensing Image Scene Classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022, doi: 10.1109/TGRS.2022.3157671.

[49] F. Zheng, S. Lin, W. Zhou, and H. Huang, “A Lightweight Dual-Branch Swin Transformer for Remote Sensing Scene Classification,” *Remote Sens (Basel)*, vol. 15, no. 11, Jun. 2023, doi: 10.3390/rs15112865.

[50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015

[51] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” Aug. 2016

[52] Y. Hua, L. Mou, and X. X. Zhu, “Recurrently Exploring Class-wise Attention in A Hybrid Convolutional and Bidirectional LSTM Network for Multi-label Aerial Image Classification,” Jul. 2018, doi: 10.1016/j.isprsjprs.2019.01.015.

[53] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, “CNN-RNN: A Unified Framework for Multi-label Image Classification.”

[54] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang, “Learning Deep Latent Spaces for Multi-Label Classification,” Jul. 2017

[55] K. Xu, P. Deng, and H. Huang, “Vision Transformer: An Excellent Teacher for Guiding Small Networks in Remote Sensing Image Scene Classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2022.3152566.

[56] A. Hassani, S. Walton, J. Li, S. Li, and H. Shi, “Neighborhood Attention Transformer,” Apr. 2022

[57] W. Wang *et al.*, “Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions.”

[58] W. Wang *et al.*, “PVT v2: Improved Baselines with Pyramid Vision Transformer,” Jun. 2021, doi: 10.1007/s41095-022-0274-8.

[59] M. Caron *et al.*, “Emerging Properties in Self-Supervised Vision Transformers,” Apr. 2021

[60] C. Li *et al.*, “Efficient Self-supervised Vision Transformers for Representation Learning,” Jun. 2021

[61] C.-F. Chen, Q. Fan, and R. Panda, “CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification,” Mar. 2021

[62] D. Zhou *et al.*, “DeepViT: Towards Deeper Vision Transformer,” Mar. 2021

[63] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” Dec. 2020

[64] L. Yuan *et al.*, “Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet,” Jan. 2021

[65] J. Pan *et al.*, “EdgeViTs: Competing Light-weight CNNs on Mobile Devices with Vision Transformers,” May 2022

[66] Y. Hua, L. Mou, and X. X. Zhu, “Relation Network for Multi-label Aerial Image Classification,” Jul. 2019, doi: 10.1109/TGRS.2019.2963364.

[67] X. Tan, Z. Xiao, J. Zhu, Q. Wan, K. Wang, and D. Li, “Transformer-Driven Semantic Relation Inference for Multilabel Classification of High-Resolution Remote Sensing Images,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 15, pp. 1884–1901, 2022, doi: 10.1109/JSTARS.2022.3145042.

[68] H. Wu, C. Xu, and H. Liu, “S-MAT: Semantic-Driven Masked Attention Transformer for Multi-Label Aerial Image Classification,” *Sensors*, vol. 22, no. 14, Jul. 2022, doi: 10.3390/s22145433.

[69] B. Ma *et al.*, “Label-Driven Graph Convolutional Network for Multilabel Remote Sensing Image Classification,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 17, pp. 2245–2255, 2024, doi: 10.1109/JSTARS.2023.3344106.

[70] J. Llamas, P. M. Lerones, R. Medina, E. Zalama, and J. Gómez-García-Bermejo, “Classification of architectural heritage images using deep learning techniques,” *Applied Sciences (Switzerland)*, vol. 7, no. 10, Sep. 2017, doi: 10.3390/app7100992.

[71] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision.”

[72] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” Jan. 2022

[73] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[74] The PyTorch Foundation, “BCEWITHLOGITSLOSS,” 2022.

[75] T. Ridnik *et al.*, “Asymmetric Loss For Multi-Label Classification.”

[76] X. Wang, L. Duan, and C. Ning, “Global context-based multilevel feature fusion networks for multilabel remote sensing image scene classification,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 14, pp. 11179–11196, 2021, doi: 10.1109/JSTARS.2021.3122464.

[77] I. Puleko, O. Svintsytska, V. Chumakevych, V. Ptashnyk, and Y. Polishchuk, “The Scalar Metric of Classification Algorithm Choice in Machine Learning Problems Based on the Scheme of Nonlinear Compromises,” 2022.

[78] L. N. Smith and N. Topin, “Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates,” Aug. 2017

[79] Pytorch, “ReduceLROnPlateau — PyTorch 1.9.0 documentation.” Accessed: Jul. 13, 2024.

[80] A. Al-Kababji, F. Bensaali, and S. P. Dakua, “Scheduling Techniques for Liver Segmentation: ReduceLROnPlateau Vs OneCycleLR,” Feb. 2022

[81] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014

[82] Q. Gao, T. Long, and Z. Zhou, “Mineral identification based on natural feature-oriented image processing and multi-label image classification,” *Expert Syst Appl*, vol. 238, Mar. 2024, doi: 10.1016/j.eswa.2023.122111.