FireSight – Wildfire Detection through UAV Aerial Image Classification

Amrita Palaparthi
Stanford University
735 Campus Drive, Stanford, CA
amritapv@stanford.edu

Sharmila Reddy Nangi
Stanford University
735 Campus Drive, Stanford, CA
srnangi@stanford.edu

Abstract

As wildfire management becomes an ever-more prominent issue, fire detection using data from aerial drones can serve as a method for early identification and tracking of wildfire spread. We leverage Vision Transformers to perform image classification on overhead photographs of fires captured by Unmanned Aerial Vehicles, focusing on the identification of wildfire presence in image data. In this vein, we explore the impacts of data augmentation on classification accuracy, the relative performance of Transformers against CNN models on wildfire classification, and the tradeoff between model size and effectiveness; we then present an ensemble model of CNN and Vision Transformer architectures that achieves 82.28% accuracy on our binary classification task.

1. Introduction

Over the past several years, increasingly devastating wildfires have posed a growing threat around the world. In California alone, over 8800 fires burned approximately 2.5 million acres of land in 2021, causing unprecedented economic and safety risks. Early detection and prediction of wildfire spread is crucial to mitigate their impact and control post-fire effects. Sensor-based tracking technologies have recently been deployed for monitoring multiple environmental stressors, resulting in immense amounts of useful information; unmanned drones, in particular, can serve as a rich source of visual data, as they are capable of capturing detailed aerial images of actively burning areas. In this project, we aim to use the vast image data collected through these devices to determine the presence of wildfires. We leverage state-of-the-art image classification techniques on predicting wildfire presence from aerial-view photographs, comparing the effectiveness of transformer models against traditional CNN-based architectures in tackling the problem of wildfire detection. We additionally explore model ensembling and data augmentation as methods to increase the effectiveness of our classification techniques, and we describe the tradeoffs between model size and accuracy through network compression.

1.1. Problem Statement

We tackle the problem of predicting whether a wildfire is present or absent in an aerial-view image taken by an Unmanned Aerial Vehicle, or UAV. Our approach considers this problem as a classification task, where the two classes predicted by our model are "Fire" and "No Fire". In this vein, our model and baseline take in JPG color images from over burning and non-burning locations in forested areas (included in the FLAME dataset) as input. The output of our model consists of binary classification scores. We examine classification accuracy and F1 score as the primary metrics of evaluation for our approach, both overall and within individual classes, and we additionally evaluate model efficiency as measured by the size of a model in MB.

2. Related Work

2.1. Capturing Wildfires from UAV Data

Previous work on wildfire detection has primarily leveraged two forms of aerial image data: satellite imagery and, increasingly, photographs captured by unmanned aerial vehicles (UAVs). Shamsoshoara et al. focus on the latter form of data in their compiled FLAME dataset [10]. FLAME consists of images taken by drones of controlled burns in Arizona forests and is described in further detail in 4. Shamsoshoara et al. frame wildfire detection in two ways: image segmentation into burning and non-burning areas and an image classification problem in which images are classified as either containing or not containing fire. The authors also evaluate performance on these images, using the Xception network architecture for classification (as introduced by Chollet [2]). Following the Xception network, they include an output layer with a sigmoid activation function to compute two class scores (see 3.1 for more details). When trained on FLAME, the Xception model achieves a 76% accuracy on the fire/no-fire classification task. In our work, we utilize FLAME's formulation of wildfire detec-

Figure 1. Sample images from both classes in the FLAME dataset

tion as a binary classification problem, aiming to assign each image in the dataset to either a "fire" or "no fire" class. We also expand on the initial classification accuracy results achieved by Shamsoshoara et al. as a baseline for our proposed transformer-based method.

In their work on EmergencyNet [7], Kyrkou et al. create a more efficient architecture suitable for on-board detection of wildfires and other disasters from UAVs. Running deep learning models on UAVs remains a challenge because these drones often lack access to large amounts of computational power. Kyrkou et al. propose the use Atrous Convolutional Feature Fusion (ACFF) to reduce the number of parameters needed without compromising on model effectiveness. These dilated convolution blocks increase receptive field by transforming images at varying dilation rates; several small dilated filters are computed separately and merged using fusion schemes including max and concatenation operators. EmergencyNet replaces traditional convolution blocks with ACFF in all layers except the first and final convolutions(see 13). This model achieves results comparable to those of fine-tuned ResNet50, VGG16, and Xception models on a dataset of images containing natural disasters (an F1-score of 95.7%), but it uses only 0.368MB of memory in comparison with the over 85MB consumed by Xception. We compare our model against EmergencyNet, examining the potential tradeoffs of model efficiency and classification accuracy with our proposed approach.

2.2. Leveraging Transformers for Aerial Image Data

Multiple recent approaches have worked to tackle CV tasks on UAV data using Transformer architectures. Notably. Ghali et al. [4] build off FLAME's Xception network to propose the use of Transformers for wildfire segmentation. In particular, the most accurate segmentation model they present is TransUnet (a hybrid transformer-CNN model based on U-Net). TransUnet uses a pretrained Vision Transformer as well as ResNet50 extract feature maps from image data. It then passes these feature maps through a decoder consisting of cascading up-sampler blocks, performing feature concatenation with the output of ResNet50 at each block. This architecture is displayed in 15 and achieves accuracy and F1-scores of 99.90%.

While Ghali et al. do propose the use of several alternate Deep CNN-based architectures for image classification, they do not use a Transformer-based architecture for binary classification. Rather, they put forward a method for ensemble learning with a backbone consisting of DenseNet and EfficientNet-B5 models. DenseNet (proposed by Huang et al [6]) is a CNN in which features derived from each convolutional layer are passed into all subsequent layers as inputs, while the architecture of EfficientNet [12] is centered around uniformly scaling CNN network width and depth in order to reduce model size while maintaining or improving performance. Ghali et al.'s ensembling approach reveals that these two networks learn complementary and diversified output feature maps, which, when combined to predict class scores, can boost classification accuracy.

Both models are first used to extract feature maps from UAV images; these features are then concatenated and passed through a sigmoid output layer to produce a classification result. This approach results in state-of-the-art classification results, with an overall accuracy of 85.12%. As Ghali et al. demonstrate the effectiveness of pretrained ViT models as part of this TransUNet architecture to create feature maps for wildfire identification, we pursue the use of these models in wildfire classification rather than segmentation. While we reference the presented ensembling strategy as a basis for our proposed model, we also note that a simpler voting-based scheme has also been applied by Minetto et al. [8] in order to combine the outputs of multiple classifiers into a single result for UAV image data, and so we compare the performance of a feature concatenation-based approach to a voting-based ensemble in our work.

Our approach to wildfire classification is influenced by Bazi et al.'s scene-classification method on remote sensing data [1]. This method builds off of the Vision Transformer architecture to classify a UAV-photographed image based on the type of location it depicts, such as an Airfield, Beach, or game space (sports arena). ViT is pretrained on Imagenet-21k and then fine-tuned on Imagenet-1k. Then, it is adapted to the task of scene classification by fine-tuning on remote sensing data for 30 iterations. To increase training data diversity, Bazi et al. employ a variety of data augmentation techniques and compare their impacts on performance; these techniques include a standard set of rotation, flipping, and brightness and color modifications, as well as Cutout, CutMix, and a hybrid approach characterized by randomly selecting any of the three aforementioned techniques. Their results show the hybrid method of data augmentation provides the greatest gain in classification accuracy across multiple remote sensing datasets, reaching 93.82% on average. Due to the nature of the wildfire classification problem, some of the presented forms of data augmentation like Cutout may remove sections of the image containing fire and therefore would not be compatible with our new task. Steiner et. al's work on training Vi-

sion Transformers for transfer learning [11] demonstrates that similar gains in accuracy from leveraging data augmentation generalize to domains beyond scene classification and techniques beyond Bazi et al.'s proposed combination of approaches, as the paper leverages RandAugment and Mixup as alternatives to this hybrid approach. Bazi et al. also explore methods in increasing efficiency by pruning layers from their ViT model prior to evaluation; their experiments reveal that classification accuracy does, indeed, increase along with network depth, though the relative gain in accuracy tapers off after 6 layers. Our approach similarly leverages a pretrained ViT model, and we adapt a hybrid data augmentation strategy with multiple techniques to boost performance.

3. Methods

3.1. Convolutional Neural Networks

As a baseline experiment, we started with using Convolutional Neural Networks for training our classification model. Following the FLAME dataset paper [10], we trained an Xception [2] network, proposed by Google-Keras from scratch. Xception model is a deep CNN (Figure 14), which is similar to the Inception network, but replacing the standard Inception modules with depth-wise separable convolutions. Each hidden layers is a pair of 2-Dimensional (2D) convolutional blocks with a size of 8 and a stride of 2 × 2. Each block follows a batch normalization and a Rectified Linear Unit (ReLU) activation function. Sigmoid activation is used in the final output layer to predict the Fire/ No-Fire probability distribution. The end-to-end model is trained on Binary Cross-Entropy Loss with Adam optimizer for 40 epochs. [1].

As a next step, we moved to using deep pre-trained Convolutional networks like ResNet and DenseNet that produced state-of-the-art image classification results. ResNet's [5] biggest innovation lies in the usage of residual blocks, that passes the input directly to the output information in each block. This simplifies the model learning as it has to train and learn for the difference, rather than the entire output. It has been shows the ResNet improves the information transition through the deep networks and speeds up the training process. DenseNet [6] is built on top of ResNet, where the input to each layer is the concatenated output from all the previous layers, to maximize the information transmission across different layers. Both these networks were pre-trained on the large ImageNet dataset ((14 million images, 21,843 classes) and our goal is to exploit the information learnt by these models. We used these models as a backbone to extract the image features and trained a linear classifier layer on top of it.

[1] https://github.com/AlirezaShamsoshoara/Fire-Detection-UAV-Aerial-Image-Classification-Segmentation-UnmannedAerialVehicle

3.2. Vision Transformer

While transformers revolutionized NLP, over the past few years, they have been extensively applied to generate state-of-the-art results in diverse domains including computer vision. This motivated us to apply transformers for wildfire classification. Vision Transformer is an end-to-end architecture that converts an input image to a sequence of patches encodes it through an embedding layer followed by passing it through the different transformer encoder layers and finally through a classification layer that predicts the score for 2 classes. Vision Transformers are powerful as they use the technique of multi-headed self attention that can capture meaningful representation of the image by determining the relative importance of a single patch with respect to all the other patches in the sequence. We leverage the Vision Transformer (ViT-Base) model with 12 transformer layers [3] by Google-Brain, pre-trained on ImageNet-21k, and we fine-tune it on the FLAME dataset for Fire/No Fire binary classification. The architecture for ViT is included in Figure 16.

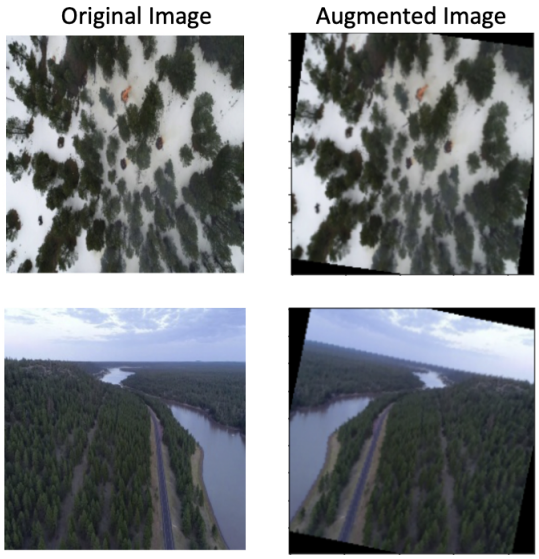

Figure 2. Examples of Data Augmentation

3.3. Data Augmentation

Deep networks (both CNN and transformers) are data-intensive and their performance is greatly benefited in the presence of large annotated training data. Data augmentation is a simple but effective strategy to increase the size and diversity of the training dataset. We uses different manipulation techniques on the training data to generate additional training samples from the existing one while preserving the validity of the original class label. Training a model on augmented data helps to combat the overfitting problem and

thus improve the robustness and the generalization ability of the model. In our project, we applied a combination of geometric transforms while training our models including horizontal flip (with probability 0.5) and rotation (15 degrees) (See Figure 5). Note that we could not apply crop / cut augmentations as there is a possibility for occasional cropping of the fire in the images,which may modify the true image labels in a non-deterministic fashion. We also explored the application of color-space transformations by adjusting brightness, contrast and saturation using color jitter and grayscale transformations.

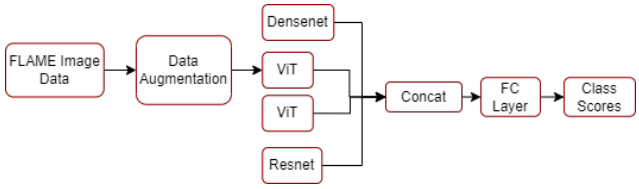

Figure 3. Ensembling Strategy

3.4. Ensemble Learning

We explore three techniques for model ensembling, examining their performance using both Vision Transformers alone and a combination of Transformer and CNN architectures:

- **Voting**: This ensembling scheme first retrieves the predicted class for each input model. It then performs a simple majority vote, selecting the most commonly predicted result as the output class of the ensemble.
- **Confidence**: The confidence-based approach first computes the softmax probabilities of both classes for each input model. The prediction associated with the highest normalized class score is then selected as the predicted score. This emphasizes models that are most confident in their predictions, rather than giving each model an equal ability to determine classification output as in the voting-based scheme.
- **Feature Concatenation**: Building off of Ghali et al. [4], this approach applies a linear classifier to the output feature maps of the input models. In this scheme, the linear output layers are removed from each input model once fine-tuning is complete, and the results from the new final layer are concatenated and passed through a fully connected layer to produce the fused class scores. This approach enables the ensemble model to learn appropriate weights for features produced by each model. See Figure 3 for a diagram of this ensemble approach for our most performant combination of models (DenseNet, ResNet, ViT, and ViT with data augmentation).

We evaluate each of these three approaches on a variety of model combinations, including solely CNNs (DenseNet and ResNet), solely Vision Transformer models, and a mixed group of both CNN and Transformer models, incorporating both models that had been exposed to augmented datasets during fine-tuning and models that had not.

3.5. Model Compression

Transformers are huge models with millions of parameters - the ViT-base model we used had 86 million parameters. They owe their best performance to these huge parameters and training data. However, this adds immense computational complexity and memory requirements which makes it difficult for practical purposes, especially, when we want to deploy it on UAVs for the real-time fire detection. Hence, we explore the direction of model compression, where we tried to study the trade-off between the model parameters and its performance to create lighter model for fire-classification. Visual transformers are also known for their redundant architecture with 12 encoder layers. We experimented with a simple pruning technique where we train separate classifier models with the features from different transformer layers. The idea is to find the early layers whose features could capture important information effectively, which enables us to prune the rest of the layers, resulting in smaller models.

4. Dataset and Features

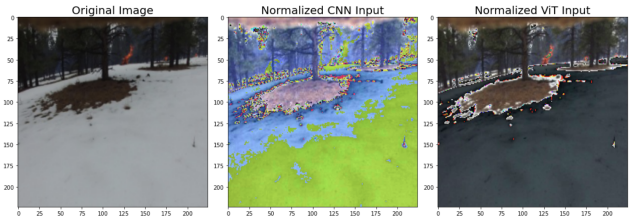

Figure 4. Original and Normalized FLAME Images

We run binary classification on the Fire Luminosity and Airbone-based Machine Learning Evaluation (FLAME) dataset flame. This dataset consists of aerial images collected by drones during a prescribed burn of piled detritus in an Arizona pine forest. The captured data includes image frames from drone-recorded videos, which are annotated and labelled with both segmentation maps and class labels based on the presence of fire. The 47,992 images of the FLAME dataset are divided into a training dataset of 39,375 images and a test set of 8,617 images (see Table 1 for details). We further set aside 10% of the training data as our validation set for hyperparameter tuning. To promote generalizability across UAV hardware, the test set and

training set of FLAME use images from different models of drones; this discrepancy results in substantially higher validation accuracy than test accuracy in both FLAME's classification baseline and our own work. All images are pre-processed to a resolution of 224 X 224 for the task of "Fire-vs-NoFire" image classification and are available for reference in IEEE dataport[2].

Subset	Class	Number of Images
Train + Validation	Fire	**25018**
	No Fire	**14357**
Test	Fire	**5137**
	No Fire	**3480**

Table 1. Train/Test Split and Class Distribution in FLAME Data

Prior to passing these images through our models, we first apply model-specific normalization transforms. All PyTorch CNN models pre-trained on ImageNet use the following normalization coefficients, which we employ for training linear classifiers upon ResNet and DenseNet architectures: $[0.485, 0.456, 0.406], [0.229, 0.224, 0.225]$. However, our pretrained ViT model uses a uniform normalization coefficient value of 0.5. We also preprocess our data with various forms of data augmentation including rotation and horizontal flip, described in detail in 3.3. See Figure 4 for a visualization of normalized CNN and ViT inputs.

5. Experimental Results

5.1. Training and Hyperparameter Selection

During training, we conducted a grid search on learning rate for both our CNN models and ViT and compared the performance of the Adam and SGD optimizers. For ViT (with and without data augmentation), our final configuration included a learning rate of $2e-4$ and an adam optimizer. For DenseNet, ResNet, and our model compression experiments using ViT with a smaller number of layers, we used an SGD optimizer with a momentum of 0.9 and a learning rate of 0.001. To reduce the impact of overfitting, we also computed the validation accuracy after each epoch of training and selected the highest-performing number of training epochs for each model. ViT models were trained with a batch size of 16, while CNN models used a batch size of 8. All models were trained with the Binary Cross entropy loss function, depicted below for n examples, predictions $\hat{Y}_i$, and ground truth labels Y_i. All our code, including our choice of hyperparameters, is available at https://github.com/ampalaparthi/firesight.

[2]https://ieee-dataport.org/open-access/flame-dataset-aerial-imagery-pile-burn-detection-using-drones-uavs

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log (1 - \hat{Y}_i))$$

Figure 5. Binary Cross-Entropy Loss

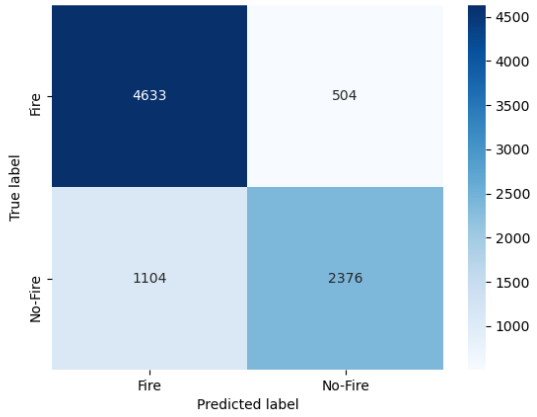

Figure 6. Confusion Matrix on Test Dataset with ViT + DA

Evaluation Metrics: We evaluate the performance of our approach using Classification accuracy and F1 score on the test set. F1 score, computed as the Harmonic mean of Precision and Recall, is a better metric to evaluate classification models. We also use confusion metrics to visualise False positives and False Negatives.

5.2. Single-model Results

Table 2 presents binary classification results on the validation and test sets on all our CNN-based and Transformer-based architectures.

Model	Validation		Test	
	Accuracy	F1	Accuracy	F1
Xception	97.00	0.96	49.09	0.58
DenseNet	98.09	0.97	70.35	0.53
ResNet	97.69	0.97	73.20	0.61
ViT	99.95	0.99	78.25	0.68
ViT+DA	99.85	0.99	81.35	**0.75**
Ensemble-Vote	99.92	1.00	81.94	**0.75**
Ensemble-Conf	99.95	1.00	81.40	0.74
Ensemble-Concat	**100.0**	**1.00**	**82.28**	**0.75**

Table 2. Single-model and Ensemble Results

We note that the pre-trained CNN models(DenseNet and ResNet) perform better than the Xception network, which is trained from scratch. This gives a signal that these models learn to understand important signals when trained on large datasets. Additionally, ViT significantly outperform

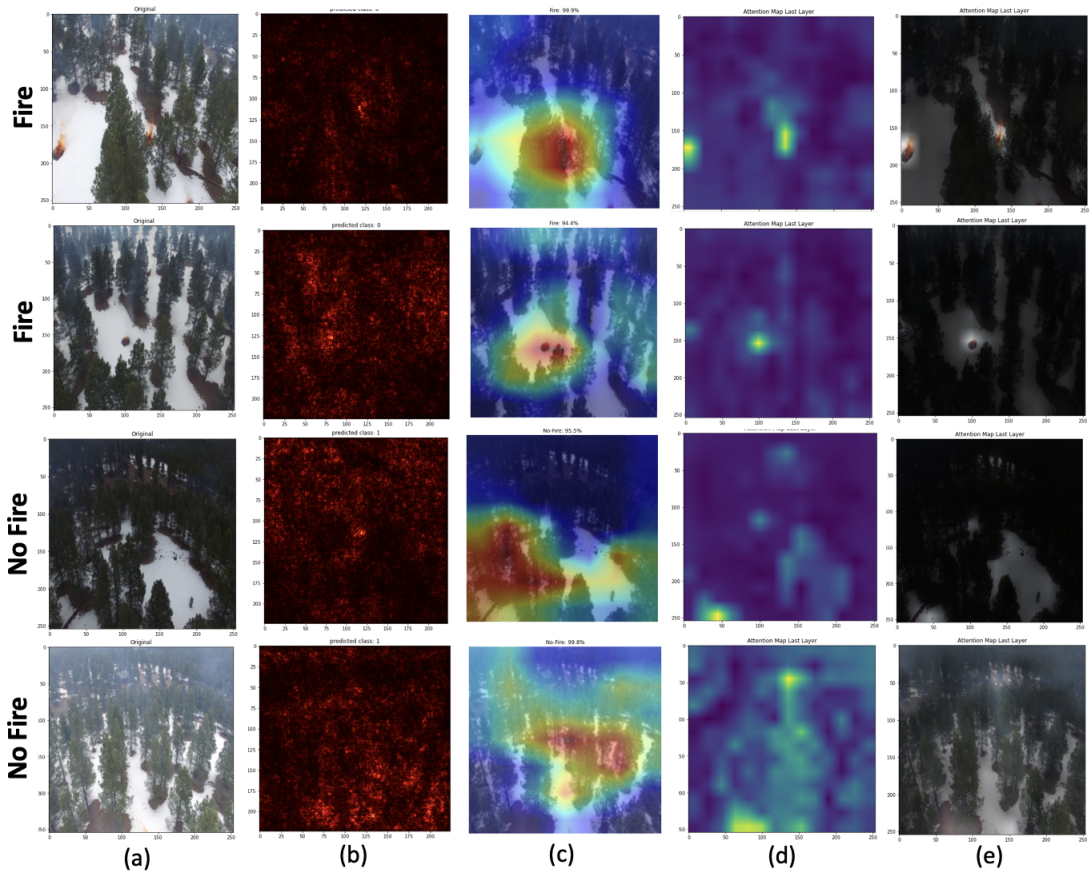

Figure 7. Qualitative visualisation of the (a) original input image with (b)Saliency Map (c) Grad-CAM from ResNet and Attention Weight visualisation from ViT(d,e). These examples are where the predicted output by the model matches with the ground truth label.

all the CNN based networks, which can be reasoned due to the self-attention mechanism, huge model parameters and the knowledge attained by pre-training ViT on ImageNet data. However, in all these models we notice that there is a large gap between the validation and the test accuracies. This is because of the fact that the dataset for train and test were collected from different drones [10], leading to a shift in the data distribution. Since validation is a small split from the training data, the models learn signals that resemble better with the validation data as against the test data. While working with the domain shift is challenging, on a brighter note, we believe that improvement of performance in test set is now, a better indicator of the model generalisation. We thus, compare the performance of the models only on the test set and not on the validation set.

Data Augmentation: We use the proposed plan for data augmentation to generate additional training data on-the-fly, while training our ViT models. We noticed that data augmentation(ViT+DA) led to a prominent improvement in the test performance from 79.25% to 81.35% in accuracy and 0.68 to 0.75 in F1-scores. This is an indicator that the training data variability has helped in model regularisation and generalise better to the test set. We found that transformations in the color space (namely color jitter and grayscale) reduced classification accuracy to approximately 60%. This suggests that color is an important factor used by our models when predicting the presence of fires, and that differentiating between fire and other dark areas of an image (such as open ground) becomes a substantially more difficult task when color is not used as a signal.

Furthermore, Figure 6 also presents the Confusion Matrix on the test set with ViT+DA model, which reveals that around 1100 Non-Fire images in the test set are misclassified as Fire (False Positive) and 504 Fires were not detected (False Negative). We believe that this use-case of Fire Detection system can tolerate False Positives, but should be stricter with False Negatives as missing the early fire detection would be a problem with huge after-effects. Our model supports this with the relatively smaller False Negative rate.

5.3. Ensemble Performance

As Table 2 delineates, our highest performing ensembling technique was the Feature Concatenation approach. The following are the combinations of models that yielded

the highest classification accuracy for each of the three methods:

1. Voting: ViT, ViT+data augmentation, and ResNet
2. Confidence: ViT, ViT+data augmentation, and ResNet
3. Feature Concatenation: ViT, ViT with DA, ResNet, & DenseNet

In all three cases, using both CNN and Transformer models as inputs improved classification accuracy substantially, boosting the performance of the Feature Concatenation approach by over 2%. This indicates that the features learned by ResNet/DenseNet and ViT may be complementary, and that a learnable combination of the two types of models' outputs can help boost performance without unduly favoring one model's results over the other; instead, the use of feature maps rather than class scores to create an ensemble model provides greater flexibility to emphasize certain valuable features output by ViT and de-emphasize others.

5.4. Qualitative Analysis

In Figure 7, we provide qualitative visualizations to help shed light on our models' decision-making processes. These include Saliency Maps for ResNet gradients, Grad-CAM [9] (Gradient-weighted Class Activation Mapping) visualizations for ResNet, maps of attention weights for ViT, and ViT's attention weights superimposed on the original input image (where brighter regions indicate larger weights during classification). The images for which "Fire" is predicted demonstrate that CNN architectures like ResNet tend to place emphasis on slightly different regions of input images than does ViT; while attention weights for these images are concentrated quite heavily on the small regions of the image containing flames, the corresponding saliency maps tend to emphasize a substantially larger region of the image including open space surrounding the fire. For images categorized as "No Fire," both ViT and ResNet adopt a more holistic approach. Though both attention weights and CNN gradients are scattered more evenly throughout these images, the areas of emphasis still differ; for instance, in the third visualization, ResNet tends to focus on the forested areas of the image while ViT highlights the open snow.

5.5. Failure Cases

As figure 8 shows, false positives associated tend to occur when a patch of open ground is misinterpreted as a fire. These open patches are sometimes misclassified as fire because they are similar in shape, size, and coloration to flames in the training dataset; to remedy this, we would need to train our model on additional data including more fires of varying sizes to supplement the relatively uniformly-shaped controlled burns in FLAME.

False negatives, where images containing fire are misclassified as "No Fire" images (see Figure 9), occur primarily in two cases: when the image was presented at a non-overhead angle as in the first example, and when large amounts of smoke were present without a corresponding fire. In the former category, fire was photographed when a UAV had not yet reached its maximum altitude, and so the model was faced with the unfamiliar task of classifying fire when seen head-on as opposed to from a bird's-eye view. Smoke also tended to be a signal of the presence of fire, especially for ResNet and DenseNet, and so including more images with smoke in our training would help remedy this association.

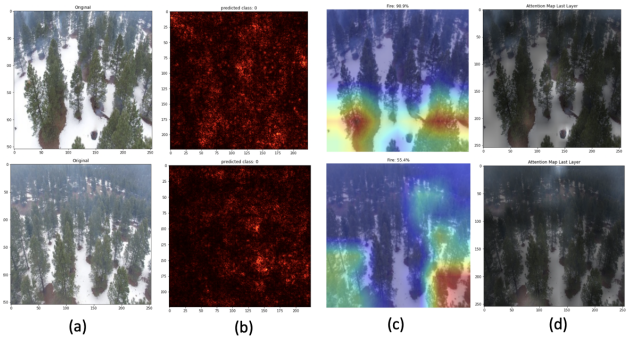

Figure 8. False Positives with Non-Fire images predicted as Fire; (a) Original Image; (b) Saliency Map and (c) Grad-CAM from ResNet (d) ViT Attention Map

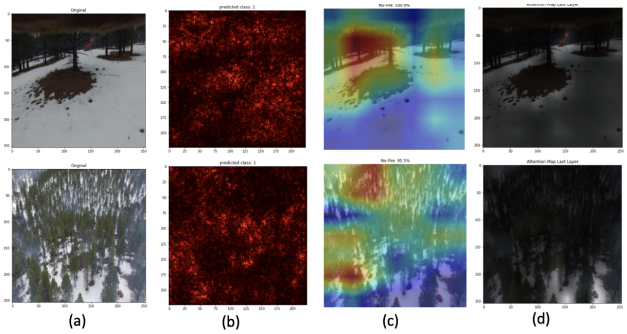

Figure 9. False Negatives with Fire images predicted as Non-Fire; (a) Original Image; (b) Saliency Map and (c) Grad-CAM from ResNet(d) ViT Attention Map

5.6. Model Compression Results

To further the set of model compression experiments, we analysed the role of different layers in the ViT. At first we trained the entire 12 layers ViT-Base and visualised the attention weights for different layers. Figure 10 presents the attention weights mapped over the input image (with fire) for different layers. Notice that the first layer does not focus on anything significant to fire/no-fire classification, but with more layers, the weights keep concentrating on the

presence of fire in the image. We can also notice that the change in concentration is significant from layer 1 to 5, but more or less remains the same after that. This hints that the transformer architecture might be learning redundant information with more parameters and can be compressed to a smaller lighter model.

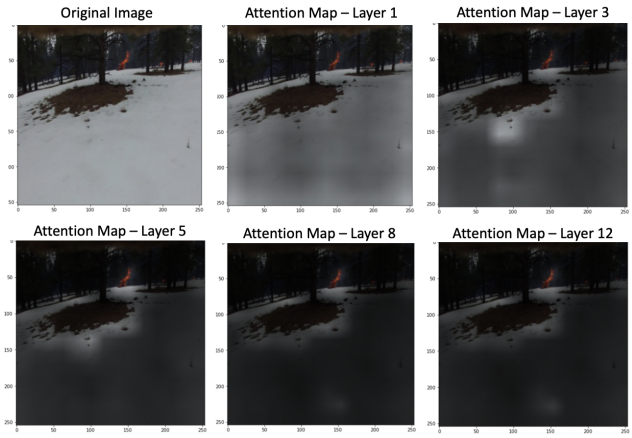

Figure 10. Attention Weight Visualisation across ViT layers

As a next step, we extract the features from different layers and train them with a classifier head. For instance, if we train with the first layer features, all the other 11 layers can be pruned from the ViT architecture. We repeated this experiment for different layers in ViT and Figure 11 presents test accuracies for different layers. Notice that there is a huge drop in accuracy when only the first layer is used, but the performance with 6 initial layers is closer to that of the full model (12 layers), reaching 76.63%. On the contrary, when we calculate the model size in terms of disk space (see Figure 12), the model with 6 layers is almost half the size (169 MB) as against the ViT-Base model (337MB). This is an interesting result as it suggests that we can reduce the model size by almost 50%, without having to compromise extremely on the model performance.

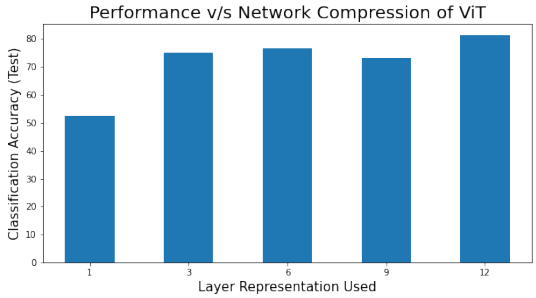

Figure 11. Performance trade-off with ViT Network Compression

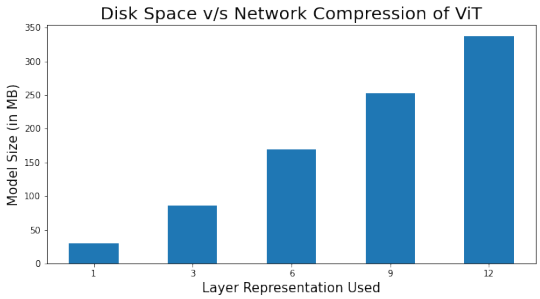

Figure 12. Disk Space trade-off with ViT Network Compression

6. Conclusion

In this report, we explore the application of Vision Transformer models to the task of wildfire classification on aerial image data, examining the impact of data augmentation, ensembling techniques, and model compression. We present an ensemble of two fine-tuned ViT models (with and without applying data augmentation), a ResNet model, and a DenseNet model that trains a linear classifier on the concatenated feature maps generated by all four networks, yielding a classification accuracy of 82.28%. We find that this approach enables the combined use of complementary features from CNN and Transformer models, weighting key output features in a learnable way in order to boost performance. Our results indicate that rotation and horizontal flip were the forms of data augmentation that resulted in the greatest gains in model effectiveness to an accuracy rate of 81.35% on a single ViT model. In addition, we see that the tradeoff between model size and classification accuracy diminishes when over 6 layers of ViT are used before the final linear classifier - comparable classification results of 76.63% can be achieved with substantially reduced memory consumption when only the first 6 layers of the model are included.

Future work that builds upon our model would consist of further compression through knowledge distillation in order to create a smaller network that can run onboard low-memory UAV devices, as well as incorporating additional training data from a variety of environments as opposed to a single forested location in Arizona. Next steps also include testing our model on images from larger, uncontrolled wildfires rather than controlled burns as well as on real-time video data from UAVs. This testing would enable us to determine if our approach performs well on applications like wildfire perimeter tracking, where changes in classification from ”fire” to ”non-fire” would indicate the position of the wildfire’s edge.

7. Contributions and Acknowledgements

7.1. Contributions

In this project, both Amrita and Sharmila contributed equally to the codebase and writeup, often working jointly on code sections. Amrita focused on fine-tuning models and model ensembling techiques, while Sharmila concentrated on data augmentation and model compression. Both worked on generating visualizations and quantitative results, and both members collaborated to write this report and the accompanying poster.

7.2. Acknowledgements

We leveraged pretrained models from HuggingFace for ViT and Torchvision for ResNet and DenseNet, and some of our code for importing and training models builds off of the corresponding PyTorch and HuggingFace tutorials. Our code for saliency map visualization expands on our solutions for CS 231N Assignment 2.

We would like to thank our project mentor Agrim Gupta for all of his guidance throughout our experience planning, implementing, and delivering this project, as well as the coordinators of the Big Earth Hackathon Wildland Fire Challenge for their support and feedback.

References

[1] Yakoub Bazi, Laila Bashmal, Mohamad M. Al Rahhal, Reham Al Dayil, and Naif Al Ajlan. Vision transformers for remote sensing image classification. *Remote Sensing*, 13(3):516, Feb 2021. 2

[2] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 1, 3

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[4] Rafik Ghali, Moulay A. Akhloufi, and Wided Souidene Mseddi. Deep learning and transformer approaches for uav-based wildfire detection and segmentation. *Sensors*, 22(5):1977, Jan 2022. 2, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2016. 2, 3

[7] Christos Kyrkou and Theocharis Theocharides. Emergencynet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:1687–1699, 2020. 2

[8] Rodrigo Minetto, Mauricio Pamplona Segundo, and Sudeep Sarkar. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6530–6541, sep 2019. 2

[9] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019. 7

[10] Alireza Shamsoshoara, Fatemeh Afghah, Abolfazl Razi, Liming Zheng, Peter Z. Fulé, and Erik Blasch. Aerial imagery pile burn detection using deep learning: the FLAME dataset. *CoRR*, abs/2012.14036, 2020. 1, 3, 6

[11] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. 2021. 3

[12] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. 2019. 2

8. Appendix

TABLE III
EMERGENCYNET MODEL STRUCTURE

Layer	Output Size	Receptive Field	Number of Filters	Stride
Input image	240 × 240			
Convolution	120 × 120	3	16	2
ACFF Block 1	120 × 120	3, 5, 7	64	1
MaxPooling 1	60 × 60	2		2
ACFF Block 2	60 × 60	3, 5, 7	96	1
MaxPooling 2	30 × 30	2		2
ACFF Block 3	30 × 30	3, 5, 7	128	1
MaxPooling 3	15 × 15	2		2
ACFF Block 4	15 × 15	3, 5, 7	128	1
ACFF Block 5	15 × 15	3, 5, 7	128	1
ACFF Block 6	15 × 15	3, 5, 7	256	1
Convolution	15 × 15	1	5	1
Global Pooling	5			
Softmax	5		5	

Figure 13. EmergencyNet Architecture

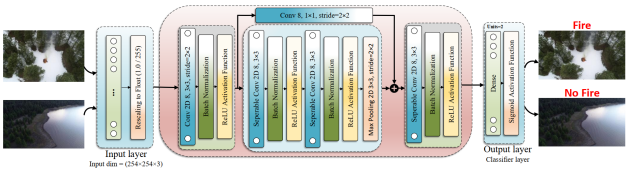

Figure 14. Xception Model Architecture for Fire Classification

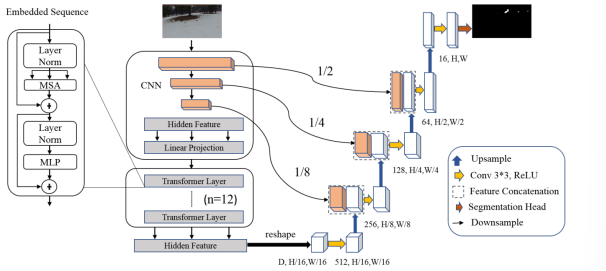

Figure 15. TransUnet Architecture

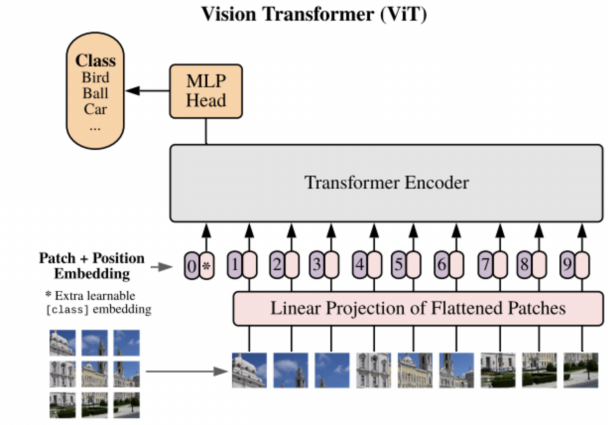

Figure 16. Visual Transformer Architecture