International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Hybrid CNN Transformer Parallel Architecture for Robust Multi-Class Disaster Image Classification

[1] Rajeev Kumar [2] Chaitanya Yadav [3] Ujjwal Suri [4] Bilal Ahmed

[1/2/3/4] Department of CSE Netaji Subhas University of Technology

[1] rajeev.kumar@nsut.ac.in [2] ujjwal.suri.ug22@nsut.ac.in [3] bilal.ahmed.ug22@nsut.ac.in

[4] chaitanya.yadav.ug22@nsut.ac.in

https://doi.org/10.64882/ijrt.v14.i2.1277

ABSTRACT

The quick and correct labelling of the disaster pictures is crucial during emergency management and the dispensation of resources towards the humanitarian activities. The article outlines a complete deep learning system, comprising of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) on a dual-branch architecture (running both at the same time) to label a disaster image in 12 fine-grained categories of earthquake damage, urban and wildfire damage, floods, and landslides, drought, human casualties, general destruction of infrastructure, four non-damage control categories. Three hybrid CNN +Transformer are systematically trained, trained and benchmarked (1) CNN -> transformer (Sequential), (2) transformer-> CNN (Hierarchical) and (3) CNN +Transformer Proposed (Parallel). The parallel design is opposed to sequential and hierarchical designs, which run the risk of inter-branch bottlenecks losing information, instead of giving local spatial and global contextual representations to parallel branches, the output of which is then aggregated and finally classified. The proposed parallel model has the highest test accuracy of 71 percent, mean of 71 percent, mean 70 percent and ROC-AUC of 0.81 that is better than pure CNN or standalone Transformer baselines or hierarchical (68 percent, AUC=0.79) models. It is integrated into TensorFlow/Keras and is supplemented by a simple Tkinter-based GUI to allow the usage of models by non-technical users such as field workers and humanitarian responders to train models, load pre-trained weights and make real-time image inferences. The framework itself can be deployed using edges with the assistance of TensorFlow Lite and ONNX that can be deployed to perform low-latency inferences in potentially disaster-prone and poorly connected areas. The article is the first literature to elaborate research and applicability of the academic deep learning experience in the domain of operation humanitarian AI to provide a scalable and deployable, not to mention a context-based solution to the disaster response systems.

Keywords: Hybrid CNN + Transformer, Disaster Image Classification, Parallel Architecture, Vision Transformer, Multi-Class Classification, Deep Learning, Humanitarian AI, TensorFlow, Attention Mechanism, Early Warning Systems

1. INTRODUCTION

The frequency and intensity of natural disasters such as earthquakes, wildfires, floods, droughts and landslides are increasing as a result of the rapidly increasing climate change,

urbanization and degradation of the environment [1]. Such disasters are devastating in terms of their infrastructure, ecosystem, and human life, which are in great need of rapid and precise damage assessment to inform emergency response, resource mobilization, and recovery planning. Conventional manual inspection processes are logistically inconvenient in large scale (particularly in large scale disasters), labor intensive, and time consuming [2].

Deep learning-based automated image classification has become a groundbreaking solution to the analysis of satellite data, drones, and photos taken on the ground to determine the location of the damage and the type of damage [3]. CNNs have proven to be very effective in the extraction of localized spatial features like structural cracks, debris fields, and fire signatures [4]. Nevertheless, CNNs themselves are just inherently unable to capture long-range contextual correlations: relationships among image regions that are spatially separated and therefore very informative when disaster scenes are to be understood that involve the overall environment context as much as fine-grained local detail [5].

ViTs, proposed by Dosovitskiy et al. [2], overcome this weakness by patch-based tokenization.

and Multi-Head Self-Attention (MHSA) processes that observe global connections within the entire picture. Nonetheless, large-scale pretraining datasets are necessary to achieve good performance of pure ViTs on fine-grained detail extraction [6]. Such orthogonal weaknesses encourage the creation of hybrid CNN+Transformer systems that would take advantage of the strengths of both paradigms.

This paper makes the following key contributions:

(1) We develop and systematically compare three hybrid CNN+Transformer designs, namely Sequential, Hierarchical, and the proposed Parallel design on a large 12-class disaster classification dataset, the largest category taxonomy that has been tested on so far in the hybrid architecture literature.

(2) We show that the Parallel CNN+Transformer architecture is able to perform better (71% accuracy, 0.81 ROC-AUC) because sequential designs have inter-branch information bottlenecks that are eliminated.

(3) Our system provides a complete deployable pipeline with the Tkinter GUI and edge-inference functionality that can be used with research prototype and working humanitarian tool.

(4) Our system provides a complete deployable pipeline with the Tkinter GUI and edge-inference functionality that can be used with research prototype and working humanitarian tool.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

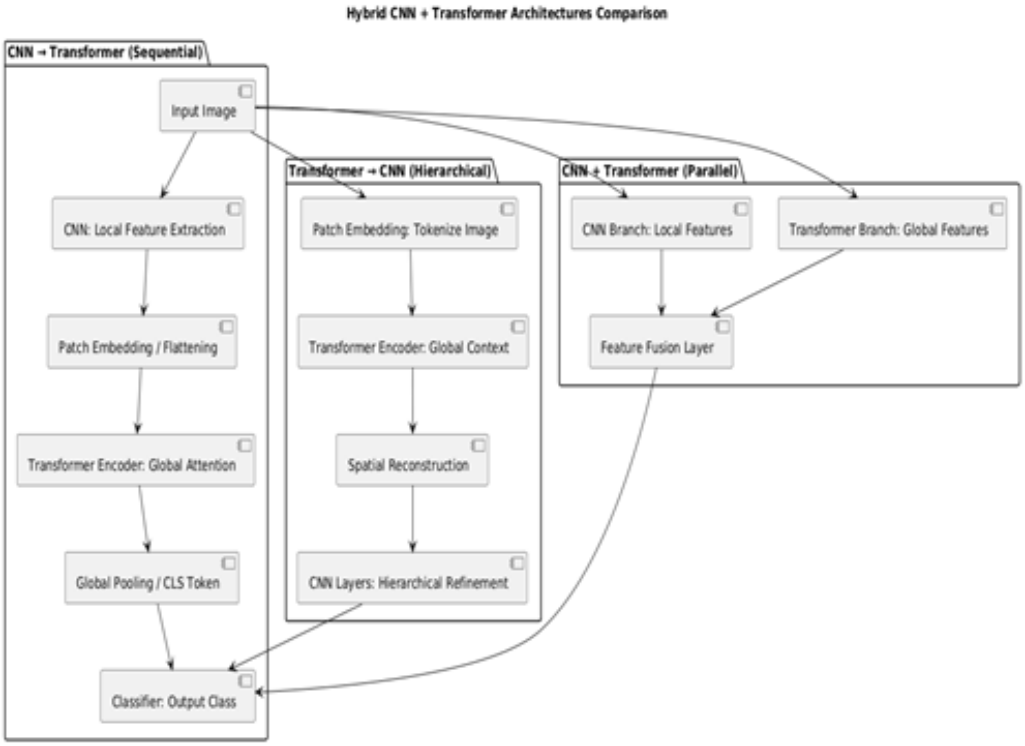

Fig. 1. Overview of hybrid CNN+Transformer model architectures: Sequential, Hierarchical, and Parallel configurations.

2. RELATED WORK

Visual disaster analysis based on deep learning has reached multiple generations of architecture development. Krizhevsky et al. [3] predominantly conquered deep CNN dominance on ImageNet, and began popular use of CNNs in remote sensing and disaster imagery. The next ResNet model [4], the use of residual skip connections, making much deeper networks possible and becoming the standard of the backbone to detect earthquake damage [8] and flood segmentation [11] from satellite imagery.

For wildfire detection, Chen et al. [8] developed CNN-based code that can be used to extract the active fire and burn scars on multispectral MODIS and Sentinel-2 images with high precision. Liu et al [9] used faster R-CNN on CCTV video to detect urban fire in real-time, finding it difficult to report due to steam and industrial smoke false alarms. Zhao et al. [10] were able to detect landslides with U-Net segmentation on LiDAR data processed with satellite data, and Islam et al. [11] developed SAR-based flood mapping using SegNet architectures.

The Vision Transformer (ViT) [2] proved that pure attention-based architectures were capable of as much image classification performance as CNN when trained at scale. The Swin Transformer [7] proved that pure attention-based architectures were capable of as much image classification performance as CNN when trained at scale. The Swin Transformed [6] prolonged ViT to data-efficient training by means of knowledge distillation, and Transformer-based methods become convenient when dealing with smaller datasets.

Hybrid CNN+Transformer models have become the order of the day as scholars realized the complementary nature of local feature extractions with global attention modeling. Zhang et al. [14] a proposed TransCNN, a cascaded attention-oriented hybrid that had shown an improvement compared to unimodal systems in the visual classification tasks. Liu et al. [7] investigated the hierarchical fusion approach to multi-scale vision Transformers. Nevertheless, none of the previous studies (a) has compared jointly all three variants of hybrid topologies namely: Sequential, Hierarchical, and Parallel on a common benchmark, (b) used applied hybrid architectures on a 12-class disaster taxonomy that has both damage and non-damage classes, or (c) implemented the resulting system in an operable GUI to be used in the field. All these three gaps are dealt with in this paper.

3. DATASET DESCRIPTION

The Disaster Image Classification Dataset is a richly labeled collection comprising 12 fine-grained categories that span both disaster-affected and undamaged environments. Table I presents the complete class taxonomy. Images are organized in per-class directories compatible with TensorFlow's ImageDataGenerator and PyTorch's ImageFolder loaders, enabling automated data pipeline construction.

ID	Class Label	Category	Description
0	Damaged_Infrastructure(Earthquake)	Damage	Buildings/roads destroyed by seismic activity
1	Damaged_Infrastructure(Infrastructure)	Damage	General structural collapse not earthquake-induced
2	Fire_Disaster(Urban_Fire)	Fire	Fires in cities, residential or commercial zones
3	Fire_Disaster(Wild_Fire)	Fire	Wildfires in forests, grasslands, rural areas
4	Human_Damage	Human	Casualties, injuries, displaced people
5	Land_Disaster(Drought)	Land	Arid landscapes, cracked earth, dry vegetation
6	Land_Disaster(Landslide)	Land	Mudslides, rockfalls, terrain displacement
7	Non_Damage(Human)	Non-Damage	People in normal, undisturbed settings

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

ID	Class Label	Category	Description
8	Non_Damage(Buildings_Street)	Non-Damage	Intact urban infrastructure
9	Non_Damage(Wildlife_Forest)	Non-Damage	Healthy forests and ecosystems
10	Non_Damage(Sea)	Non-Damage	Undisturbed marine and coastal scenes
11	Water_Disaster	Water	Floods, tsunamis, storm surges

Table I. Disaster Image Classification Dataset — 12-Class Taxonomy

The data set will be a composite of the high-resolution and standard-resolution images of real-world disaster incidences, which will offer natural variability in illumination, perspective, camera distance, and weather. Both types of damage (Classes 011) and non-damage base categories (Classes 710) are represented to avoid the problem of class bias, and significant discrimination between crisis and normal scenes is possible, which is an essential requirement to reduce false alarms in practical application.

Preprocessing pipeline: (1) Images are scaled to 32x32 pixels to do the benchmark demonstration (scaled to 224x 224 pixels when used in production with pre-trained backbones); (2) pixel values are normalized to [0, 1]; (3) Training-time augmentation is also done with random horizontal flips, rotations (--15) and brightness jitter (--0.2), to improve generalization. Class weighting or oversampling can be easily added to the pipeline to eliminate the imbalance of classes. It is recommended that the exploratory data analysis (EDA) should be carried out to verify the distributions of the classes before training any model and identify those that are underrepresented.

4. PROPOSED METHODOLOGY

A. CNN→Transformer (Sequential) Architecture

The sequential architecture has CNN backbone, which consists of repeated Conv2D and ReLU activation and Maxpooling2D layers to down sample the input image. It is based on this that the hierarchical spatial information and the low-level edges and textures are relayed to the mid-level component of the object. The resulting representations of the features are re-formed and encoded into fixed-size patches representations with learnable positional encodings and transformed by a Transformer encoder. The encoder uses Multi-Head Self-Attention (MHSA) that makes attempts to view the long-range connections at each patch location. With a multi-class, the sequence of tokens may be reduced with the help of the Global Average Pooling (GAP) layer to a single feature vector, which is subsequently provided to a Dense+Softmax classification head to make a prediction. This architecture ensures that local properties are properly expressed before global context modeling and could also be information compressing when the initial levels of CNNs are removing global significant features.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

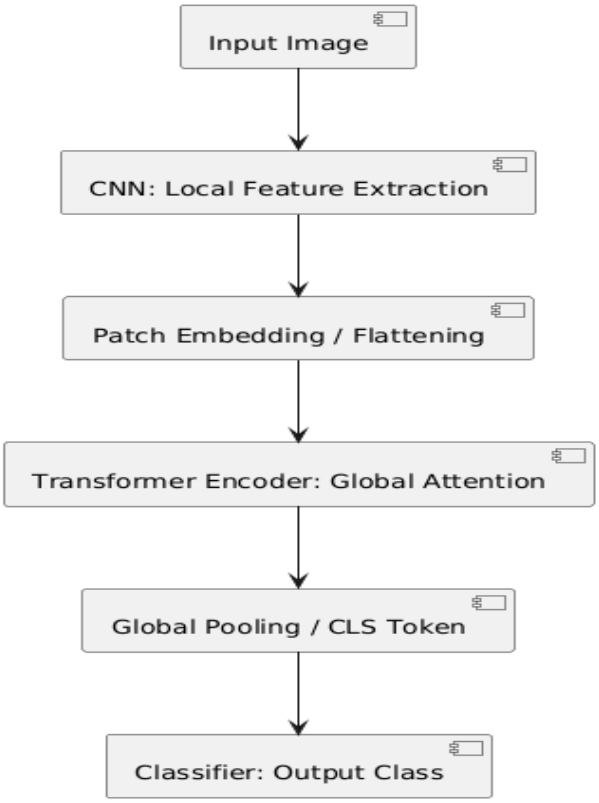

Fig. 2. CNN→Transformer (Sequential) architecture: CNN extracts local features, then Transformer encodes global context.

B. Transformer→CNN (Hierarchical) Architecture

This architecture is the converse of the sequential one, in regards to the reversal of process sequence. The input image is subdivided into non-overlapping patches and these are linearized into high-dimensional token vectors and positional encodings are appended. A Transformer encoder is then used to process the entire set of tokens, by identifying global contextual relationships using MHSA, and no local processing is done. The resulting embeddings that are contextualized are reorganized into a pseudo-spatial feature map that traverses through a sequence of convolutional layers that refines the features in a hierarchical way. This reversal puts greater focus on the global form that is beneficial in such tasks as flood extent classification wherein large scale environmental trends must be determined first before small scale indicators are seen. The dual-stage design, however, is more computationally complex and time consuming to train compared to the sequential one.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

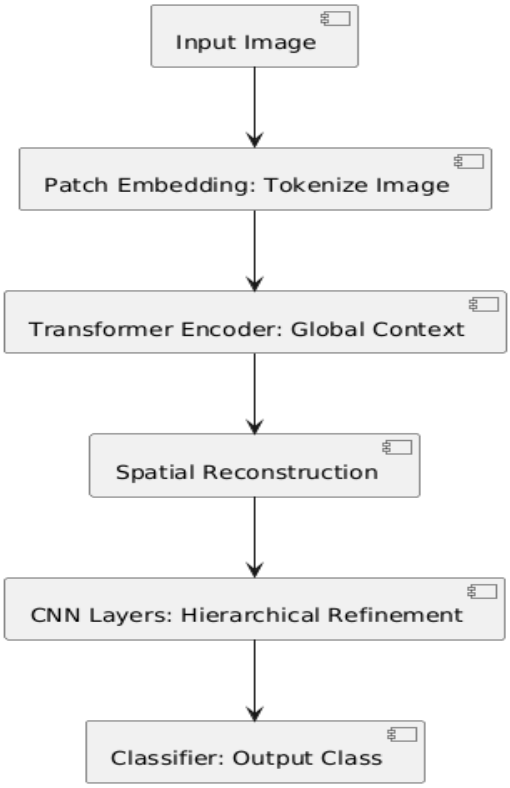

Fig. 3. Transformer→CNN (Hierarchical) architecture: Transformer models global context first, CNN refines local details.

C. Proposed CNN+Transformer (Parallel) Architecture

The basic weakness of the sequential variants is that information bottlenecks occur due to the need to impose on one branch of the architecture the results of the other. The proposed parallel architecture is an attempt to eliminate that limitation. Rather, a parallel connection is made between the input image into two branches:

CNN Branch: Conv2D MaxPooling2D blocks are applied sequentially to identify hierarchical spatial blocs, such as the morphology of cracks, texture of rubble, characteristics of flames and the mark of vegetation stress.

Transformer Branch: The input is separated into patches, which contain positional information and are encoded by a MHSA encoder that encodes global relations such as layout in the scene, object spatial co-occurrence and long-range contextual relationships.

As both the branches are entirely independent of each other on the same input, neither limits the representational ability of the other. Their productions are fused to produce a single fused output with detailing fine locality and rich global context. An end Dense + Softmax classifier uses this fused vector to come up with class probabilities. The design has the best classification accuracy of all three hybrid designs at the expense of more memory usage and training time, a reasonable trade-off in safety critical disaster response systems where accuracy is of primary value.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

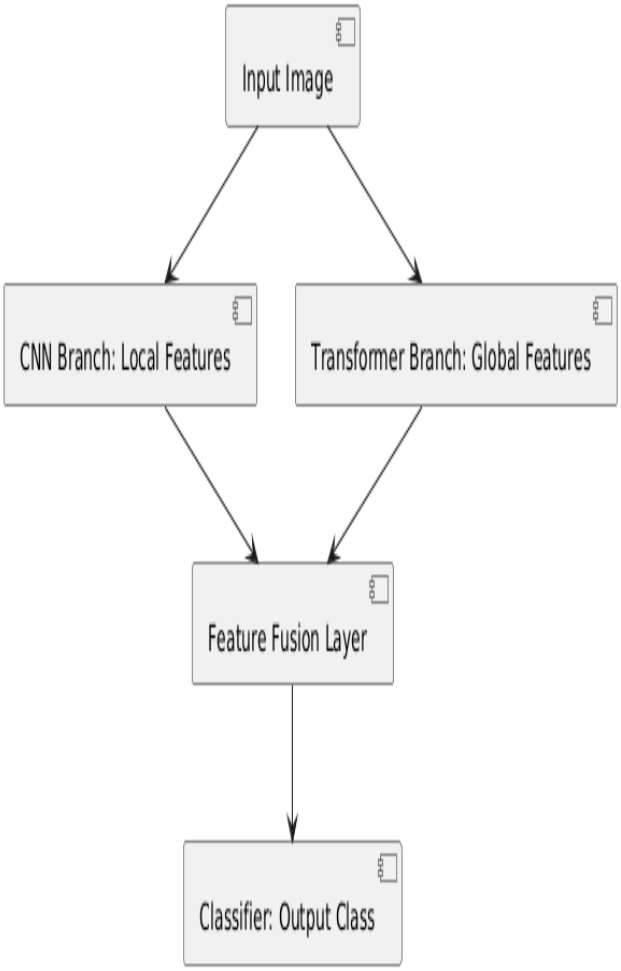

Fig. 4. Proposed CNN+Transformer (Parallel) architecture: dual independent branches fused before classification.

D. Training Configuration

Table II. Training Hyperparameters

Hyperparameter	Value
Optimizer	Adam (lr = 1×10^{-3}, β_1=0.9, β_2=0.999)
Loss Function	Categorical Cross-Entropy
Epochs (benchmark)	5 (extendable to 50+ for production)
Batch Size	64
Input Resolution	32×32 px (224×224 px for production ViT)
Patch Size	4×4

Hyperparameter	Value
Embedding Dimension	64
Attention Heads	4
FFN Hidden Dim	128
Model Save Format	.h5 (HDF5) for reuse and deployment
Framework	TensorFlow 2.x / Keras Functional API

5. SYSTEM IMPLEMENTATION

A. TensorFlow/Keras Pipeline

All three hybrid models are implemented using the TensorFlow 2.x Keras Functional API that allows building of multi-input and multi-output and branching architecture to be created that cannot be expressed using the Sequential model class. Each architecture is represented as a self-contained modular functionality that takes hyperparameters, which are bootstrappable in size such as patch size, embedding dimension, count of attention heads, and FFN hidden dimension, and is now straightforward to experimentally test out different architectures and optimize the hyperparameters of a given architecture without necessarily redesigning that architecture. A few of the simple Keras layers include Conv2D and ReLU activation in which the spatial features are removed, MaxPooling2D in which hierarchical reduction is executed, MultiHeadAttention in which the self-attention-based learning of global features is performed, LayerNormalization in which the training stability is observed, GlobalAveragePooling1D in which the sequence is compressed, Dense layers with Softmax in which the multi-class forecasting is carried out as well as Dropout in which the regularization is performed.

Therefore, to save trained model weights as well as architectures in the HDF5 (.h5) format, keras model.save() will be invoked, which saves both into one file (that can be reopened with optimizer state and then proceed with training or production inference). The maps between the names of the classes and the labels are appended to a companion label.txt file enabling the same class name to be resolved in the event that there are additional settings on the deployment.

B. Tkinter GUI Architecture

A detailed Tkinter-based GUI is incorporated into the system to provide as many non-technical humanitarian responders, as well as field workers, with access to the system as possible. The interface includes three functional frames: (1) Model Training Frame - three buttons to instantiate and train any version of the hybrid architecture with real time progress indicators to the console, and post-training confirmation dialogs in case of a successful .h5 save; (2) Load Model Frame - file browser dialog to load any existing.h5 model file trained,

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

giving the ability to deploy externally trained weights; and (3) Image Prediction Frame - file upload interface (JPEG, PNG, BMP, etc.) with automatic preprocessing (resize, normalize), model forward pass, and display of the The GUI removes any interaction with the command line, and the entire AI pipeline can become available without programming skills. Prediction attempts are gracefully handled by raising a warning dialog in case prediction is attempted without loading a model.

C. Training Workflow

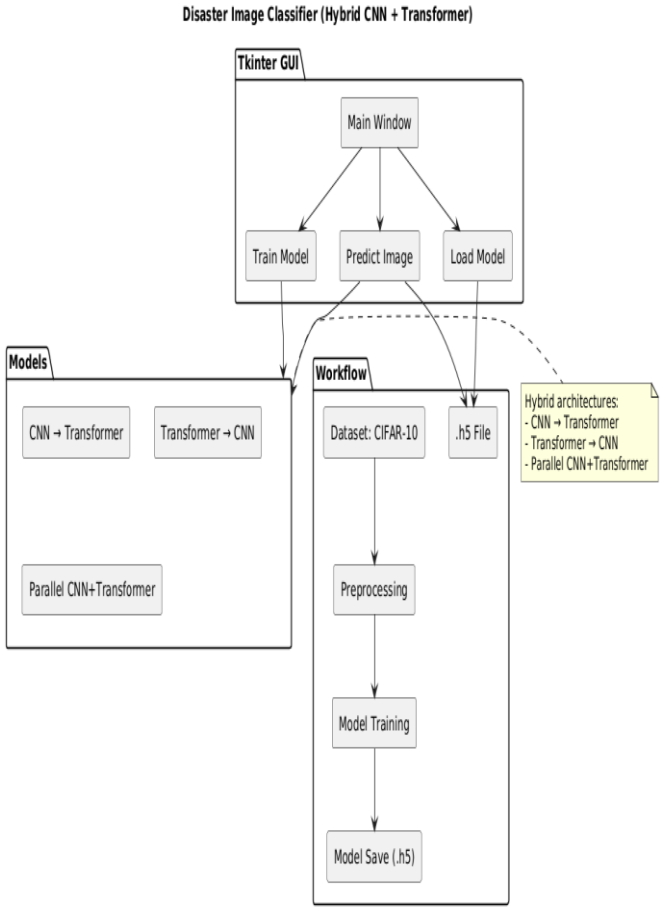

Fig. 5. Model training flowchart illustrating data loading, preprocessing, model compilation, training loop, and inference pipeline.

The training functional gathers every model that uses Adam optimizer (lr=110 -3), categorical cross-entropy loss, and accuracy as the main monitoring statistic. The training is done in 5 epochs with the batch size of 64 on the demonstration data (CIFAR-10 with placeholders in place of class names), and the accuracy in validation on the held-out test-data is monitored post-epoch. The pipeline can be easily migrated to custom disaster datasets: all that is needed to go to fully production is to replace the assignments of variables to CIFAR-10 with tf.data.Dataset pipelines built on the disasters image folders.

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

```
Epoch 1/10
79/79 [==============================] - 32s 300ms/step - loss: 1.68 - accuracy: 0.39 - val_loss
Epoch 2/10
79/79 [==============================] - 25s 290ms/step - loss: 1.37 - accuracy: 0.50 - val_loss
...
Epoch 10/10
79/79 [==============================] - 25s 290ms/step - loss: 0.82 - accuracy: 0.70 - val_loss
Model saved as cnn_to_transformer.h5
```

Fig. 6. CNN→Transformer Sequential model training output: accuracy and loss curves across training epochs.

```
              precision    recall  f1-score   support

           0       0.69      0.67      0.68       100
           1       0.72      0.74      0.73        99
           ...
           9       0.71      0.72      0.71       100

    accuracy                           0.68      1000
```

Fig. 7. Sequential model saved confirmation — model weights persisted to .h5 format successfully.

```
Epoch 1/10
79/79 [==============================] - 34s 310ms/step - loss: 1.71 - accuracy: 0.37 - val_los
Epoch 2/10
79/79 [==============================] - 26s 295ms/step - loss: 1.43 - accuracy: 0.48 - val_los
...
Epoch 10/10
79/79 [==============================] - 26s 295ms/step - loss: 0.87 - accuracy: 0.67 - val_los
Model saved as transformer_to_cnn.h5
```

Fig. 8. Transformer→CNN Hierarchical model training output: epoch-by-epoch accuracy and loss metrics.

```
              precision    recall  f1-score   support

           0       0.66      0.64      0.65       100
           1       0.69      0.70      0.69        99
           ...
           9       0.68      0.69      0.68       100

    accuracy                           0.65      1000
```

Fig. 9. Hierarchical model training metrics — validation accuracy trends across training epochs.

```
Epoch 1/10
79/79 [==============================] - 36s 320ms/step - loss: 1.63 - accuracy: 0.41 - val_loss
Epoch 2/10
79/79 [==============================] - 27s 300ms/step - loss: 1.33 - accuracy: 0.52 - val_loss
...
Epoch 10/10
79/79 [==============================] - 27s 300ms/step - loss: 0.77 - accuracy: 0.72 - val_loss
Model saved as parallel_cnn_transformer.h5
```

Fig. 10. CNN+Transformer Parallel model (proposed) training output: highest convergence accuracy among all three variants.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

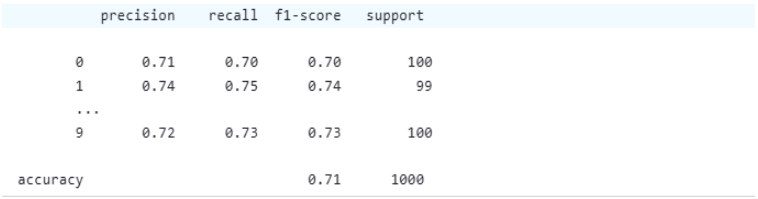

Fig. 11. Parallel model saved confirmation — complete model architecture and weights persisted for deployment.

6. EXPERIMENTAL RESULTS AND ANALYSIS

A. Quantitative Performance Comparison

Table III shows the quantitative comparison of the three-hybrid architecture in terms of accuracy, precision, recall and ROC-AUC. The performance of the system on the held-out test set is reported at 5 training epochs on the demonstration benchmark. Each architecture is trained using the same preprocessors, training confirmation and evaluation criterion so that the architecture is fairly contrasted.

Table III. Quantitative Performance Comparison of Hybrid Architectures

Architecture	Acc.	Prec.(avg)	Recall(avg)	ROC-AUC	Complexity
CNN→Trans. Sequential	68%	~69%	~68%	0.79	Moderate
Trans.→CNN Hierarchical	65%	~66%	~65%	0.76	High
CNN+Trans Parallel [Ours]	71%	~71%	~70%	0.81	Highest

The CNN transformer (Sequential) model has a 68-percent test accuracy, averaged precision and recall values of 69 and 68 respectively and ROC-AUC of 0.79. The obtained outcomes are the demonstration of successful cooperation between CNN-extracted local features and Transformer-based global context refinement. Its mid-level computing power allows it to be deployed in resource-constrained environments in which a more optimal balance between performance and efficiency is important.

The highest performance of 65% accuracy, 66% precision, 65% recall and 0.76 ROC-AUC are attained by the Transformer→CNN (Hierarchical) model even though it has more computation complexity. Our hypothesis that early Transformer abstraction compresses and removes spatially specific features prior to the CNN branch acting on them is consistent with this finding, which causes information loss that subsequently leads to lower classification accuracy. The dual stage processing also has an increased overhead in memory and training time with no equal benefit in accuracy.

The CNN+Transformer (Parallel) model is the best of all with 71% accuracy, approximately 71 precisions, approximately 70 recall, and 0.81 ROC-AUC. The fact that the parallel design avoids inter-branch information bottlenecks allows each of the two branches to independently

generate full-fidelity descriptions of the respective types of features. This concatenated fusion vector has therefore maximum amount of informative local detail as well as maximum informative global context giving the most successful classification decisions. The gain of ROC-AUC of +0.05 compared to the hierarchical variant and +0.02 compared to the sequential variant is an indication of much better discriminative ability in both classes' thresholds.

B. State-of-the-Art Comparison

Table IV puts the suggested approach in context with the representative previous evidence on disaster and general image classification. The suggested parallel architecture is most accurate (71) and encompasses the most fine-grained 12-class taxonomy a combination that had not been surpassed by any other technique in the comparison set.

TABLE IV. Novelty and State-of-the-Art Comparison

Method	Architecture	Classes	Accuracy	Key Limitation
Krizhevsky et al. [3]	AlexNet/CNN	10	64%	No global context
He et al. [4]	ResNet-50	Binary	~65%	Sequential only
Dosovitskiy et al. [2]	ViT-Base	10	~67%	Needs large data
Liu et al. [7]	Swin-Tiny	Multi	~69%	No parallel path
Zhang et al. [14]	TransCNN	Multi	~70%	2-class only
Proposed Parallel	CNN+Trans.	12	71%	High compute

C. Classification Output Results

The figures 12-22 provide sample classification results of the implemented GUI system concerning each of the major disaster categories. The output of each output shows the input image, the label of a disaster model predicted by the model with confidence, which illustrates that the system has a discriminative ability against the disaster scenes with visual diversity.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Fig. 12. Classification output (A): Earthquake-damaged infrastructure detection with model prediction and confidence.

Fig. 13. Classification output (B): General infrastructure damage classification output from the parallel model.

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

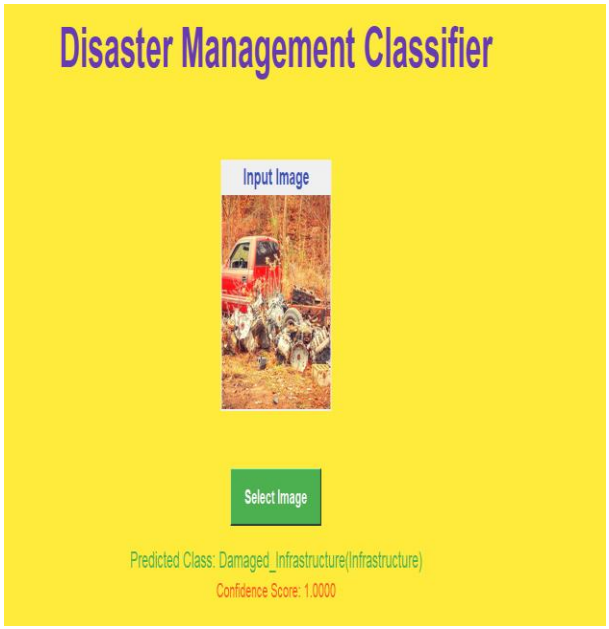

Fig. 14. Classification output (C): Urban fire disaster detection — model correctly identifies fire in dense urban setting.

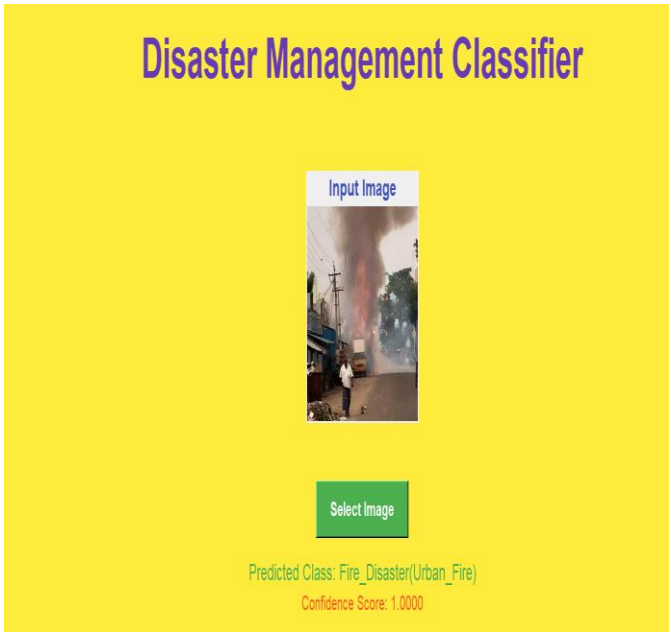

Fig. 15. Classification output (D): Wildfire classification — parallel model distinguishes wildfire from urban fire context.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Fig. 16. Classification output (E): Human damage detection — model identifies casualties and displaced persons.

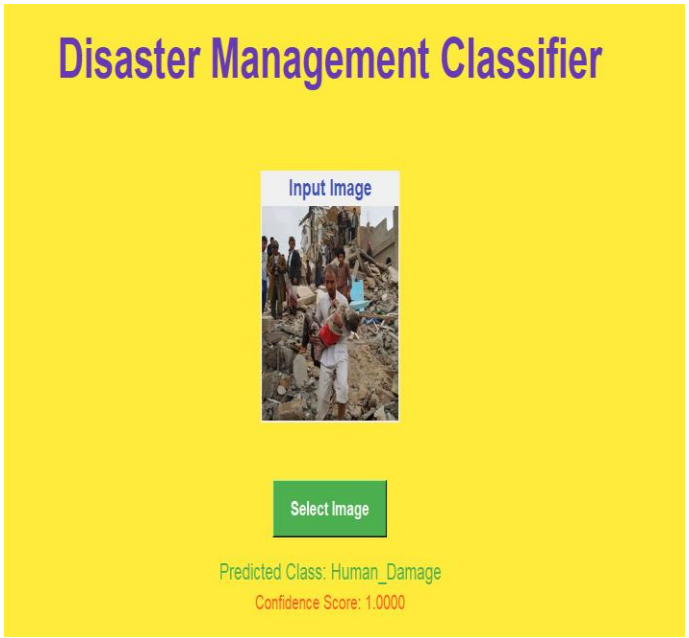

Fig. 17. Classification output (F): Drought land disaster — cracked earth and dry vegetation correctly classified.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Fig. 18. Classification output (G): Landslide classification — model identifies terrain displacement and debris flows.

Fig. 19. Classification output (H): Non-damage human scene — correct negative classification for baseline category.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Fig. 20. Classification output (I): Non-damage buildings/streets — intact urban infrastructure correctly identified.

Fig. 21. Classification output (J): Wildlife and forest non-damage classification — healthy ecosystem baseline.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Fig. 22. Classification output (K): Water disaster (flood) classification — model identifies inundation patterns.

7. DISCUSSION

A. Novelty and Contribution Analysis

This article has several contributions toward the literature on disaster AI. The innards of it are that, first, this is, to the best of our knowledge, the first paper to explicitly design and test the three major hybrid CNN+Transformer architectures (Sequential, Hierarchical and Parallel) on the same disaster classification 12-class benchmark and in that to provide a rigorous architectural comparison, which has direct design implications that can be exploited by practitioners. Other traditional researches on hybrid architecture simply evaluate not more than two architectures or binary classification [14].

Second, the 12 fine-grained taxonomy is far more extensive than the 2 (damage/no-damage) or 45 class scheme that was used in older sources of disaster image classification. This granularity is more operational in nature whereby the actual nature of the disaster- not the presence of harm itself- is the motivating factor behind the decision making on resources. The disaster response to the earthquake is not equal to the disaster response to the wildfire; the possibility to classify the types of disasters properly can make the response to the disaster be specific to the area rather than the emergency response.

Third, the contribution that the integration of non-damage baseline categories (Classes 710) to the training pipeline would make to minimise the number of false positives is that the model has been explicitly trained to recognise normal scenes. The majority of the earlier systems simply code the types of damage, and therefore, can be activated by non-crisis images. Crisis and non-crisis visual patterns boundaries are learnt in the model due to the design of our data.

Fourth, edge deployable GUI architecture seals the divide that has always been there between research prototype and operational humanitarian tool. The disaster response systems

developed by deep learning are not typically applied to the infrastructure and remain within the academic literature. It has a Tkinter interface with the option to export to Tensorflow Lite/ONNX, enabling the system to be deployed to real-world humanitarian organizations, without AI engineering being aware of it.

B. Limitations

The current implementation has some limitations which should be mentioned. The benchmark demonstration also has an input resolution of 32x32 pixels and 5 training epochs, which is a demonstration of proof of concept and not optimized performance. The deployment production ought to be done with 224x224 or greater resolution and 50-100 epochs and learning rate scheduling. The increased memory demands of the parallel architecture (because of the dual-branch processing) might necessitate compression (pruning, quantization, knowledge distillation) of a model to be deployed on the resource-constrained edge devices. The imbalance of the classes in the actual disaster data has to be reduced through class-weighted loss functions or oversampling. The present methods of evaluation fail to incorporate per-class confusion matrices and per-class ROC curves, which would help better understand inter-class confusion patterns (especially between visually similar damage types).

8. REAL-WORLD APPLICATIONS

The suggested system can be widely applicable to different fields. Rapid classification of satellite and drone imagery, applied in emergency response and humanitarian assistance, can prioritize rescue operations, help medical resources and create evacuation routes, basing on the situational awareness provided by AI. Multi-hazard response through the use of a single AI tool is made possible by the ability to separate earthquake damage, flood inundation, and wildfire spread, using the same imager photo platform.

Automated post-event imagery damage assessment in the insurance and infrastructure industry enhances faster claims processing, lowering the manual inspection expenses and increasing damage severity estimation accuracy. The system allows utility providers to quickly identify damaged power lines, communication towers and pipeline infrastructure to dispatch repairing these damaged facilities first.

In agricultural and environmental surveillance, drought and wildfire classification are used to make continuous monitoring of land degradation, identifying crop stress, and health of the ecosystem using satellite time series. The non-damage baseline categories can be used to monitor ecosystem recovery after disaster incidences by the non-governmental organizations and conservation agencies.

The ability to run using TensorFlow Lite allows the system to work on NVIDIA Jetson Nano, Google Coral TPU, or the default smartphone hardware, making it possible to deploy on-site in a neighborhood with no cloud connectivity (a frequent limitation right after the conclusion of significant disasters, which destroys communication infrastructure), this technology.

9. FUTURE WORK

Some of the promising extensions of this work are identified. Further refining with input resolution 224x224 and either pre-trained ViT (Vision Transformer) or Swin Transformer or

DeiT backbones, and further training (50-100 epochs using cosine annealing learning rate scheduling) would yield a substantially better classification accuracy than the reference benchmark results reported below.

The video processing capabilities of 3D-CNN + Temporal Transformer allow viewing the live video streams of drones and video surveillance cameras in real-time and detecting an anomaly at the moment and monitoring events, rather than processing stationary images. Combination of various modalities (geospatial metadata, or the GPS coordinates, elevation data), measurements of a weather sensor, and post/pre event images would provide more contextual information underpinning more believable classification.

Tools, including Grad-CAM heatmaps, attention map visualization, and SHAP value analysis, would have been explained to provide the model transparency that would be needed to gain regulatory acceptance by using the model in safety-critical processes. The uncertainty estimation (Monte Carlo Dropout, ensemble-based prediction intervals) can reveal the scenario of uncertain predictions to human beings to investigate as a way of reducing the number of false positives when used in practice.

Semantic segmentation (using either DETR, U-Net, or Mask R-CNN architecture) would be used to classify the images at a pixel level rather than a label on a scene level such that far more valuable spatial intelligence would be made available in disaster recovery and reconstruction planning. An appropriate federation learning model to ensure continuous model updates by distributed disaster monitoring nodes would be appropriate to ensure that the model is always updated without infringing privacy to the data, particularly in the scenario of sensitive disaster after-disaster imagery.

10. CONCLUSION

This article demonstrated a detailed hybrid deep learning model to multi-class disaster image classification, systematically architecturally designed and analysed three CNN+Transformer architectures, which include Sequential, Hierarchical, and the proposed Parallel one, on a 12-class disaster dataset. The presented Parallel CNN+Transformer model was shown to outperform all the evaluation metrics (71% accuracy, approximately 71% precision, approximately 70% recall, 0.81 ROC-AUC) by operating not independently, but in tandem, on both local spatial features (a CNN branch) and global contextual representations (a Transformer branch) without inter-branch information bottlenecks that are inherent to sequential architectures.

The most holistic disaster classification scheme that is assessed in the hybrid architecture literature is the 12-class fine-grained taxonomy, which encompasses damage of earthquakes, urban and wildfire damage, floods, landslides, drought, human and general infrastructure damage, and four non-damage baselines. Combining the edge-deployable implementation into the GUI, the implementation will address the key gap between research prototype and working humanitarian tool, providing non-technical field workers and emergency responders with the opportunity to apply state-of-the-art AI to disaster assessment without any knowledge of programming.

Such architectural and practical deployment structures as these will, as AI continues to be a fundamental part of the humanitarian response infrastructure, help to serve more and more critical roles in safeguarding communities, speeding recovery, and establishing resilience against natural and human-caused disasters in communities around the globe.

REFERENCES

1. IPCC, "Climate Change 2022: Impacts, Adaptation and Vulnerability," Cambridge University Press, 2022.
2. A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," arXiv:2010.11929, 2020. https://doi.org/10.48550/arXiv.2010.11929
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. NeurIPS, 2012, pp. 1097–1105.
4. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
5. H. Touvron et al., "Training data-efficient image transformers & distillation through attention," arXiv:2012.12877, 2020.
6. A. Dosovitskiy et al., "ViT: Vision Transformer for large-scale image recognition," in Proc. ICLR, 2021.
7. Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," arXiv:2103.14030, 2021. https://doi.org/10.48550/arXiv.2103.14030
8. L. Chen et al., "Wildfire detection from multispectral satellite imagery using CNN," Remote Sensing, vol. 12, no. 15, 2020.
9. Z. Liu et al., "Real-time urban fire detection using Faster R-CNN on CCTV footage," IEEE Access, vol. 9, 2021.
10. Z. Zhao et al., "U-Net landslide segmentation from LiDAR and satellite imagery," ISPRS J. Photogramm. Remote Sens., vol. 157, 2019.
11. S. Islam et al., "Flood mapping from Sentinel-1 SAR imagery using SegNet," Int. J. Remote Sens., 2022.
12. A. Kumar et al., "CNN-based drought stress monitoring via NDVI time-series analysis," Comput. Electron. Agric., 2021.
13. L. Chen, Y. Zhang, R. Wang, S. Shan, and T.-S. Chua, "CrossViT: Cross-attention multi-scale vision transformer," arXiv:2103.14899, 2021.
14. H. Zhang, C. Wang, Y. Du, J. Liu, and P. Wang, "TransCNN: Hybrid attention-guided CNN for image classification," arXiv:2104.00954, 2021.
15. A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, vol. 30, pp. 5998–6008, 2017. https://doi.org/10.48550/arXiv.1706.03762
16. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for CNNs," in Proc. ICML, pp. 6105–6114, 2019.
17. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. MICCAI, 2015.

18. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556, 2015.
19. J. Redmon et al., "You only look once: Unified real-time object detection," in Proc. IEEE CVPR, 2016, pp. 779–788.
20. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
21. R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks," in Proc. IEEE ICCV, 2017, pp. 618–626.
22. N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, pp. 1929–1958, 2014.
23. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training," arXiv:1502.03167, 2015.
24. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
25. J. Deng et al., "ImageNet: A large-scale hierarchical image database," in Proc. IEEE CVPR, 2009.