Aerial Fire Image Synthesis and Detection

Sandro Campos and Daniel Castro Silva
Faculty of Engineering of the University of Porto, Artificial Intelligence and Computer Science Laboratory, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

Keywords: Fire Detection, Unmanned Aerial Vehicle, Convolutional Neural Network, Data Imbalance, Data Augmentation, Generative Adversarial Network, Multi-agent System.

Abstract: Unmanned Aerial Vehicles appear as efficient platforms for fire detection and monitoring due to their low cost and flexibility features. Detecting flames and smoke from above is performed visually or by employing onboard temperature and gas concentration sensors. However, approaches based on computer vision and machine learning techniques have identified a pertinent problem of class imbalance in the fire image domain, which hinders detection performance. To represent fires visually and in an automated fashion, a residual neural network generator based on CycleGAN is implemented to perform unpaired image-to-image translation of non-fire images obtained from Bing Maps to the fire domain. Additionally, the adaptation of ERNet, a lightweight disaster classification network trained on the real fire domain, enables simulated aircraft to carry out fire detection along their trajectories. We do so under an environment comprised of a multi-agent distributed platform for aircraft and environmental disturbances, which helps tackle the previous inconvenience by accelerating artificial aerial fire imagery acquisition. The generator was tested using the metric of Fréchet Inception Distance, and qualitatively, resorting to the opinion of 122 subjects. The images were considered diverse and of good quality, particularly for the forest and urban scenarios, and their anomalies were highlighted to identify further improvements. The detector performance was evaluated in interaction with the simulation platform. It was proven to be compatible with real-time requirements, processing detection requests at around 100 ms, reaching an accuracy of 90.2% and a false positive rate of 4.5%.

1 INTRODUCTION

The extreme environmental conditions increasingly promoted by climate change make it particularly likely for natural disaster phenomena to occur each year. The especially vulnerable sub-tropical climate of the Mediterranean basin, as an example, starts outlining a trend of abnormally extended and powerful fire seasons (Turco et al., 2019). Consequently, Southern European countries such as Portugal, Spain, Italy and Greece have been frequently ravaged by uncontrolled and disproportional fires leaving trails of destruction behind (PORDATA, 2020). If not for fires, storms, droughts, and floods are amongst the many disasters that unfortunately take place. In fact, most death and damage is related to the latter three (WMO, 2021). According to the United Nations, weather-related disasters have surged five-fold in just a short time frame of 50 years, impacting poorer countries the worst. All of these are reminders of worldwide concerns that require closer coordination of means and agile mechanisms for both controlling and preventing them. This work aims to stimulate the use of fire imaging techniques to improve aerial fire detection, considering simultaneously a possible expansion to other scenarios.

The pertinence of studying these natural disasters has driven scientists to develop simulation tools capable of managing vehicles under coordinated missions. *The Platform* is an example of such a tool, and recent developments have allowed aerial vehicles to assess fire propagation by means of sensor readings (Almeida, 2017) (Damasceno, 2020); however, the potential of performing disaster control using aerial imagery is still unexplored.

Our work aims at filling the previous gap, by focusing on the development of an external module to enable the creation of a pipeline for synthetic fire generation and detection using aerial imagery. It primarily tackles the following tasks:

1. Generation of synthetic flames and smoke on aerial images captured by the simulated aircraft;
2. Adaptation of a lightweight model to detect fire in the generated images, in a real-time scenario.

Campos, S. and Silva, D.
Aerial Fire Image Synthesis and Detection.
DOI: 10.5220/0010829400003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 273-284
ISBN: 978-989-758-547-0; ISSN: 2184-433X

This work follows a recent and growing trend of producing synthetic data to train highly complex machine learning models (Tripathi et al., 2019), with applications to domains such as autonomous driving (Hollosi and Ballagi, 2019), product identification in warehouses (Wong et al., 2019), and even fire detection (Park et al., 2020). This line of thought preaches the generation of diverse and large datasets, which typically mix real samples with synthetic ones to help reduce the inconvenience of data imbalance faced by most prediction problems. The models constructed by this technique are then of use in the real domain with improved results. In this work, we address the problem of fire detection using imagery. The fire images generated by our model are assessed according to their degree of realism, both quantitatively and qualitatively, and proven to be of value by demonstrably good real fire detectors standards.

The remaining of this document is structured as follows. Section 2 provides a literature review of image generation and classification techniques, particularly adapted to the fire domain. Sections 3 and 4 present more detailed information about the proposed solution and its implementation, respectively. Section 5 describes the mechanisms used to validate the quality of the generated images and the performance of the fire detector. Finally, Section 6 gathers relevant conclusions and future work topics.

2 STATE OF THE ART

Three main strategies are primarily considered when one intends to automate the synthetic image generation necessary for the simulation of an onboard camera. The first one resorts to image rendering based on CAD (Computer-Aided Design) models, the second to compositing techniques, and the third to state-of-the-art Generative Adversarial Networks (GANs). More recently, deep-learning-based image inpainting has also proved to produce realistic features in imagery, especially when the context of their surrounding environment is considered.

Real-time optical fire detection approaches also leverage the power of deep learning models. Pretrained with extensive and diverse sets of aerial images, these models have become competitive by deploying such capabilities to devices with low computational resources.

2.1 Computer Aided Design

Computer Graphics Software (CGS) has more recently found its way into popularity due to the increasing computational power sprawl, as more capable Graphics Processing Units (GPUs) surge. Software tools such as Autodesk 3ds Max[1], Blender[2] and Unity 3D[3] enable the manipulation of CAD models, three-dimensional polygonal meshes representative of objects, and provide a suitable environment for creating virtual scenes. These applications often include rendering engines responsible for encoding the world information into a synthetic image and scripts are developed to perform batch generation. Figure 1 illustrates a fire simulation attempt using Corona Renderer[4] for Autodesk 3ds Max.

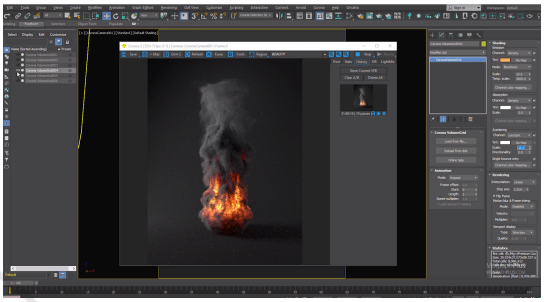

Figure 1: Three-dimensional fire rendering using Corona for Autodesk 3ds Max (Mograph Plus, 2018).

Approaches based on rendering of CAD models, despite being proven to produce realistic high-quality images, require heavy computational resources and are very dependent on human intervention for image customization and rendering (Arcidiacono, 2018).

2.2 Image Compositing

Another technique, known as image compositing, consists of extracting foreground objects from images and pasting them on new backgrounds (Rother et al., 2004). In comparison to image rendering, it is less demanding when ensuring global image high quality and local consistency.

Driven by the lack of annotated images, Dwibedi et al. presented Cut, Paste and Learn, in 2017, a method to generate synthetic images by applying this concept while focusing on patch-level realism (Dwibedi et al., 2017). This advancement was significant, considering placing features over image backgrounds may create pixel artifacts at a local level, which, when propagated into the neural classifier, may induce it to ignore the introduced features, failing their detection.

[1] Available at: https://www.autodesk.com/products/3ds-max/overview

[2] Available at: https://www.blender.org/

[3] Available at: https://unity.com/

[4] Available at: https://corona-renderer.com/

The literature does not provide much insight into fire image synthesis using image compositing techniques. This can be explained by the difficulty of segmenting flames and smoke and obtaining viable masks for accurate overlapping.

2.3 Generative Neural Networks

First proposed by Ian Goodfellow, generative adversarial networks are deep generative models consisting of two deep neural networks, a generator G and a discriminator D, opposing each other in a min-max zero-sum game (Goodfellow et al., 2014). The generator is responsible for creating synthetic data out of a latent vector $p_z(z)$, while the discriminator evaluates whether data is real or fake, when in comparison with real samples from the same domain. The two networks are connected, considering the output of the generator is, along with the real dataset, provided as input to the discriminator. GANs are trained to minimize the generator's error rate, until convergence is reached, with the improvement of the generator's data creation skills and the increasing inability of the discriminator for detecting the forged imagery. Figure 2 portrays a schematic representation of a GAN.

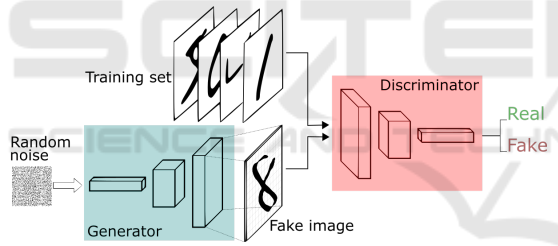

Figure 2: GANs are comprised of generator and discriminator networks. The generator produces fake data for a target domain. The discriminator provides feedback on its outcome by comparing it to real training data (Silva, 2017).

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (CycleGANs), presented in 2017, is a type of generative neural network which enables the construction of two bijective mappings, reverse of one another, between two image domains (Zhu et al., 2017). Data augmentation techniques using CycleGANs have appeared in domains where imagery is difficult or expensive to acquire, as is the case of fire detection (Park et al., 2020).

Park et al. identified the problem of class imbalance in the wildfire detection domain and presented a solution based on synthetic fire image generation (Park et al., 2020), employing CycleGANs (Zhu et al., 2017) and DenseNets (Densely Connected Convolutional Network) (Huang et al., 2017). CycleGANs enable the creation of fire images from previously collected non-fire images, by allowing the conversion of domain and the respective introduction of fire visual features. Cycle consistency and identity mapping losses are considered to prevent the model from performing unintended changes of shape and color to the original image backgrounds, while maintaining them. This procedure allowed for a better balance between image classes to be fed into the neural network. Wildfire images support increased from 43% to over 49% and allowed to almost double the total number of images on the dataset. Figure 3 presents a sample of wildfire images generated by this approach.

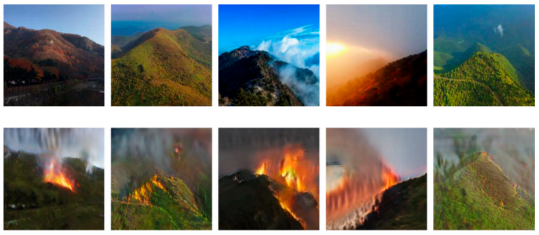

Figure 3: The images of mountains in the top row are successfully translated to the fire domain using a CycleGAN, with the respective results portrayed in the bottom row (Park et al., 2020).

2.4 Image Inpainting

Neural networks have also intervened in image inpainting, the process which focuses on restoring deteriorated images. It can include filling missing parts, repairing casual damage and removing unintended artifacts such as noise, scratches and other distortions (El Harrouss et al., 2020). These techniques aim to leave no trace of reconstruction to increase image realism and make tampering as undetectable as possible. As a consequence, they are also considered for introducing new features into imagery.

Liu et al. proposed a novel approach with a generation phase subdivided into rough and refinement sub-networks combined with a feature patch discriminator, as seen in Fig. 4.

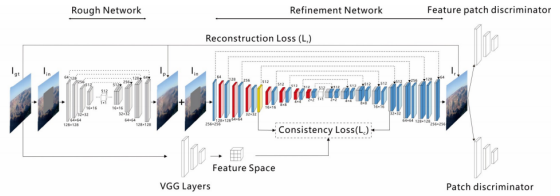

Figure 4: Architecture of the Coherent Semantic Attention network. The first sub-network creates rough pixel predictions while the second refines them to obtain better correlation of pixels between patches (Liu et al., 2019).

The rough network predicts initial rough features for unknown patches based on known neighbouring regions, advocating global semantic consistency. These are afterwards refined, in the sub-network where an auxiliary coherent semantic attention (CSA) layer is included. It allows generated patches to have a better correlation with neighbouring patches of the same unknown region, largely increasing coherency between pixels at a local level. This layer is located at resolution 32x32, as it appears to optimize model performance and needed computing requirements. Moving the layer to shallower positions may cause loss of information and increase the computational overhead due to the operations being performed at higher resolutions, while shifting it to deeper positions enhances execution times at the expense of image quality. A pretrained VGG-16 (Simonyan and Zisserman, 2015) network is also of use to extract features from the original images, introducing them as input on downsampling layers of the refinement network to speed up and optimize feature generation.

2.5 Optical Fire Detection

Many fire detection approaches using UAVs still rely on the communication with ground stations for data processing. These stations are usually equipped with high-end computing hardware capable of executing the heaviest of prediction models. However, in reality, UAVs performing missions on disaster control are subject to very limited visibility and connectivity. As a consequence, scientists are encouraged to pursue the development of self-contained, fully autonomous embedded systems for fire detection based on lightweight implementations of state-of-the-art deep learning methods.

Kyrkou and Theocharides developed a custom CNN (Convolutional Neural Network) architecture named ERNet, for emergency response and disaster management, highlighted in this work (Kyrkou and Theocharides, 2019). Their approach opposes that of many techniques, which adapt pre-trained networks, such as that of ResNet-50 (Residual Neural Network) (He et al., 2016) and VGG-16, in a process of transfer learning for image classification, resorting to the use of high-performance GPUs. They limited the number of filters applied in order to speed up computations and reduced parameter size, according to the scarce memory available onboard of such vehicles. Residual connections on the computational blocks were also useful to improve model accuracy, while not hindering performance significantly.

This network was trained to classify disasters according to 4 different incident types, in which fire is included. AIDER (Aerial Image Dataset for Emergency Response) is the augmented dataset created for this purpose. The detector achieved a mean accuracy of 90% at 53 FPS (Frames Per Second) and it consumed no more than 300 KB of memory, allowing for onboard real-time detection and on-chip storage.

3 PROPOSED SOLUTION

The proposed solution comprises the development of an external service, designated Fire Module, capable of interacting with vehicles to perform fire generation and detection in images captured from an aerial perspective. In a real environment, considering it is embedded in the firmware of actual aircraft, and given that aircraft shall be provided with onboard cameras for image acquisition on assessment missions, this module is most helpful in performing detection. In a simulated environment, however, the solution considers the existence of a service that simulates said cameras by generating aerial imagery of a specific type of disturbance. Consequently, one may require an additional module to simulate the actual behaviour of the disturbance over the terrain.

We integrate this architecture within *The Platform*, a multi-agent distributed system that provides a simulation environment based on Microsoft Flight Simulator X (FSX) for fleets of autonomous, heterogeneous vehicles. These vehicles intervene in the process of assessing disturbances in missions that range from pollution source identification to fire detection in outdoor environments (Silva, 2011). Figure 5 depicts the relevant components of The Platform's architecture for this work and their respective relationships.

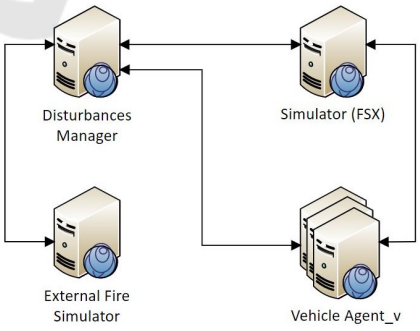

Figure 5: Relevant entities of The Platform and their respective interactions. The Disturbances Manager generates disturbances that affect the simulation environment where vehicles perform missions. The External Fire Simulator helps to reproduce the realistic behaviour of fire. (adapted from (Damasceno, 2020)).

The Vehicle Agent is responsible for the simulation of an aircraft in FSX, enabling, for instance, nav-

igation control (Silva, 2011). The Disturbances Manager (DM) creates and manages all disturbances in the simulation environment. ForeFire intervenes as an external disturbance simulator that more accurately and realistically simulates the fire spread behaviour over the terrain (Filippi et al., 2014).

The solution congregates therefore five interacting entities: the previously existing Vehicle Agents, DM and ForeFire, and the new Fire Module, comprised of a Camera Simulator and a Fire Detector, and the Maps API (Application Programming Interface), an external aerial tiles provider. The container diagram of Fig. 6 depicts the integration of the microservice within this simulation platform, the most relevant components and their relationships.

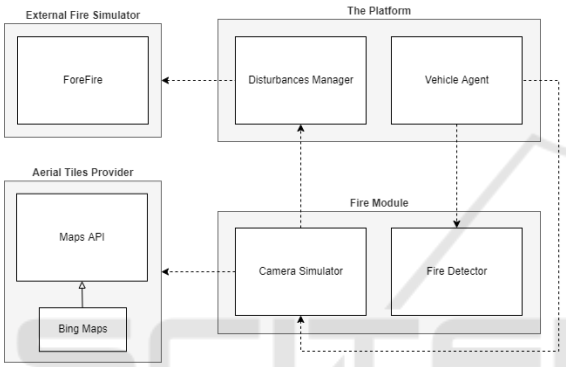

Figure 6: Container diagram depicting the Fire Module within the system and its respective dependencies.

The Fire Module has been designed as a microservice based on a RESTful architecture, as it shall provide independent, loosely coupled features to several vehicle instances cooperating simultaneously. This approach aims to increase modularity, increase resilience to faults, and ease the deployment process for devices on the edge. This decision preaches better isolation of concerns and the integration of solutions using diversifying technologies, valuing flexibility and, most importantly, the system's scalability. The module is synchronised with the DM and therefore portrays the disturbances consistently for all vehicles participating in the same mission, advocating better management of resources.

When a Vehicle Agent is performing a fire assessment mission, it connects to the Camera Simulator and requests the aerial image of its own point of view. This image, originally collected from the aerial tiles provider (ATP), at the vehicle's position, may or may not contain flames and smoke, depending on both the distance of the vehicle to the fire area and the orientation of the camera in relation to it. For that, the fires' location is always provided to the Camera Simulator by the DM. In case the simulated camera does not capture the fire, the image returned to the vehicle corresponds to the tile just as it was obtained from its provider, meaning it does not undergo any change. Otherwise, a residual neural network trained using CycleGAN proves its ability of performing non-fire to fire image domain translation, on demand, and returns the synthetic image with the fire features in place. The Vehicle Agent then provides the received image to the lightweight neural network of the module's Fire Detector which performs binary classification on the received aerial tile and evaluates the presence of fire. The aircraft requests images of the simulated camera for every position of its trajectory and repeats this process without ever knowing the ground truth of the fire detection problem.

The generation of fire by the model is highly dependent on the scenario its training has targeted. The focus in this work lies in the particular synthesis of fire for images of forests even though it can also be of use for urban environments.

4 IMPLEMENTATION DETAILS

The Fire Module provides vehicles with camera simulation and detection services. It was developed using FastAPI, a high performance tool for building APIs in Python, and its communication with The Platform is performed using HTTP. The generation service is reached using a GET request to "/camera", and its response includes the synthetic aerial image, in JPEG format, for the aircraft's position. On the other end, the detection service is reached using a POST request to "/detector", whose body shall carry an image of JPEG format as well. The API takes care of feeding the image into the ERNet classification network and returns a boolean regarding the presence of fire in it.

The camera simulator must therefore be capable of acquiring the aerial tiles corresponding to the positions of the aircraft from a preestablished maps API. Bing Maps REST Services[5] is developed by Microsoft and provides free licensing plans offering 125 thousand API requests a year and up to 50 thousand within any 24-hour period, for educational purposes (Microsoft, 2021). Apart from the traditional top-down satellite imaging, 45° angle aerial views resembling captures taken by UAVs are also available in this API and exerted much influence in its selection. Figure 7 portrays sample tiles of the two perspectives.

The CycleGAN network was trained for 110 epochs, following a learning rate of 0.0002 up to epoch 100, from which it linearly decreased, and us-

[5]More information at: https://docs.microsoft.com/en-us/bingmaps/rest-services/

Figure 7: Samples of satellite and bird's eye perspectives, as taken from Microsoft Bing Maps.

ing the Adam solver ($\beta_1 = 0.5$) as optimizer, as specified by default (Zhu et al., 2017). Also, taking advantage of the fact that the Bing Maps service uses tiles of size 256x256 pixels for rendering, the network's input size matches this value. The batch size is set to 1 to enable very frequent parameter updates and it is used with Instance Normalization layers, which are recommended for styling transfer tasks (Huang and Belongie, 2017).

The insertion of fires in the images is spatially restricted, according to the evolution of the burning area, and the procedure takes this factor into consideration. Initially, the camera simulator performs the request of the aerial image for the desired location and calculates the planar coordinates of the fire polygon (as provided by the DM) on the image. For that, an internal service of the Bing Maps API is used to draw the polygon in the appropriate location, in an easily identifiable color such as fuchsia and, as observed in Fig. 8, it is then extracted using HSV segmentation.

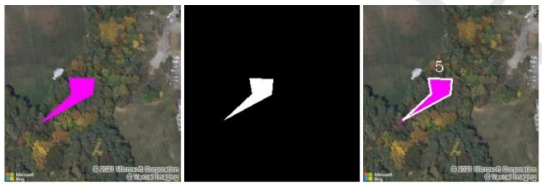

Figure 8: Polygon extraction using an edge detection technique. The generation of a binary mask allows to extract the polygonal coordinates of the designated fire region.

After translating the image collected from the ATP to the fire domain, and having the binary mask of the fire polygon, the Cut-and-Paste technique is applied and the desired result is obtained. Observe the example in Fig. 9. Note that the resulting image may or may not contain fire features, even if one is occurring, depending on the distance of the aircraft to the fire's location and the orientation of the camera in relation to it, in the case of bird's eye perspective.

In order to disguise discontinuities created by this technique, the Poisson blending method (Pérez et al., 2003) was implemented. The results generated were more realistic yet much more discrete. Figure 10

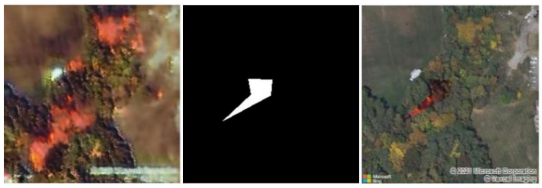

Figure 9: Example of applying the Cut-and-Paste technique for a fire polygon. The binary mask allows to select the pixels from the first image belonging to the desired fire region. They are then superimposed on the original image.

portrays an image sample with a simple superimposition of the fire polygon and the respective image when mixed seamless cloning is applied. This solution combines the gradients of the original image with those of the fire polygon to form the blended region of interest.

Figure 10: Sample image with a simple fire polygon overlay and a sample image subject to mixed seamless cloning.

At the same time, the fact that no smoke emerges from the fire polygon is odd. Looking for a solution to recreate this behaviour, we noticed that fires in confined indoor spaces are better documented as the number of variables to assess is smaller when in comparison to fires in the open. In that type of closed environment, the energy released by fires is characterised by four steps: Incipient, Growth, Fully Developed and Decay stages, as seen in Fig. 11 (Hartin, 2008).

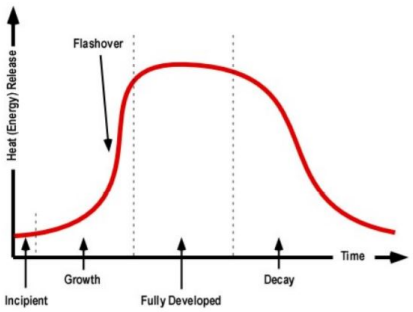

Figure 11: Stages of fire development in a compartment (Hartin, 2008).

For lack of better judgement, the method we implemented for calculating the size of the smoke columns follows a naïve approach where the energy

release, in each iteration of the fire front, is directly proportional to the size of its smoke columns. We decided that they would initially grow at a constant rate for two iterations, stay in the fully-developed phase for one iteration, and decay for over the last four iterations of the simulation, from which the fire is extinguished. Using an exclusively manual procedure, smoke vectors collected by web scraping were resized and blended into the image with the fire features. These were displayed in varying shades of grey and assuming the direction of the wind obtained directly from the simulator. Figure 12 presents a sequence of tiles representing a terrain with fire and the respective smoke progression.

Figure 12: Example of smoke progression during a simulated fire. The direction of the smoke columns is that of the wind provided by the simulator, while their sizes vary according to the respective fire stage.

Each tile represents an iteration of the generation pipeline, meaning that the speed of smoke simulation is directly proportional to the the cadence of requests made to the API. For a certain tile the smoke columns are also considered to be subject to the same wind intensity and direction. This is an approximation to what happens in reality because, as is well known, fires can tamper with local environmental conditions, and it is hardly viable to take into account the actual wind behaviour for this specific scenario.

4.1 Experiments using Coherent Semantic Attention

The image painting strategy indicates, although there is yet no scientific evidence, that there should be a possibility of filling the unknown regions of a non-fire image with flame and smoke features, as a human painter would. This assumption led to experiments that produced detailed textures and whose insertion generated little to no discontinuity. It was also observed that the application of red filters on the images directly influences the amount of flames produced, which allows to increase their variability. Figure 13 reveals some of the results. The top row depicts the original aerial images layered with a red filter and the respective fire polygons, in gray, while the bottom row depicts the same polygons filled with the flames and smoke features.

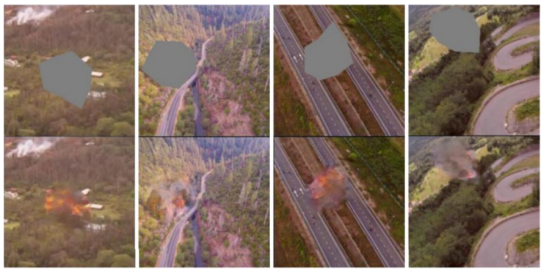

Figure 13: Samples of fire generation using CSA for images with overlay red filters and random polygonal masks. The respective results are portrayed in the bottom row.

Although appealing, this method was disregarded because the generation for large masks is unfeasible. More specifically, in this scenario, it is hardly possible to recreate the key features of the original images, which end up being stripped from their own context.

5 VALIDATION AND RESULTS

The generated fire images were evaluated according to their degree of realism, quantitatively using quality and image similarity metrics, and qualitatively by subjective and manual analysis. The adaptation of a model for fire detection, proven to be good in the real domain, also allows assessing the good performance of the synthetic generation model.

5.1 Synthetic Image Quality

The Fréchet Inception Distance (FID) allows assessing the degree of quality and similarity between the images created by the generation model and the real fire images (Heusel et al., 2017). It is based on the activations of the penultimate layer of the pre-trained InceptionV3 network and it evaluates the distance between the Gaussian distributions of the two sets of images. The FID scores, depicted in Fig. 14 for the last five training iterations, depict a minimum value of 42.0 which is part of a decreasing trend. Since the lower the FID score, the better the image quality, we may conclude that the generator is producing increas-

ingly more realistic images and with fewer artefacts (noise, blur and distortions).

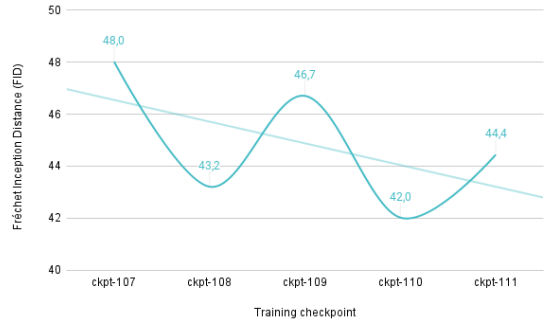

Figure 14: The Fréchet Inception Distance (FID) scores for the last five training iterations of the fire generator depict a declining trend.

Nevertheless, the most used metric to evaluate the results of the generative networks still relies on the subjective opinion of individuals, comparing real samples with fictitious ones, in Preference Judgement Surveys (Borji, 2019). For that purpose we developed a survey with a medium set of 40 generated fire images, carefully collected to hold 10 samples of forest and urban scenarios, both of top-down and bird's eye perspectives. We afterwards asked the respondents to indicate their preference in relation to the scenario and image perspective, and requested the identification of generation anomalies. Considering the relatively small population size, and for results to be robust and more representative of reality, all image samples were chosen at random for each of the previous questions. The exception to this lies on the final question, in which we decided to test the users' perception of reality. It consisted on the identification of generated samples when these were presented next to a real one, in an environment of similar configuration. The images we specifically selected for this case study are depicted in Fig. 15.

The survey was disseminated by the community and 122 responses were obtained. On a scale of 1 to 5, the subjects considered the images to have a median value (Mdn) of 4 when it comes to their degree of realism, with ratings presenting a mean value (M) of 3.44 and a standard deviation (SD) of 1.16. The generated image of forest fire approximated its real counterpart well but it was identified without much effort, managing to mislead over 12.3% of the population. The corresponding question targeting urban images deceived people similarly, with a 11.5% failure rate, but falls a bit short in its approximation to the real counterpart, exhibiting a lower M and higher SD. The population diverged more while providing their opinion on images of urban scenarios, which is

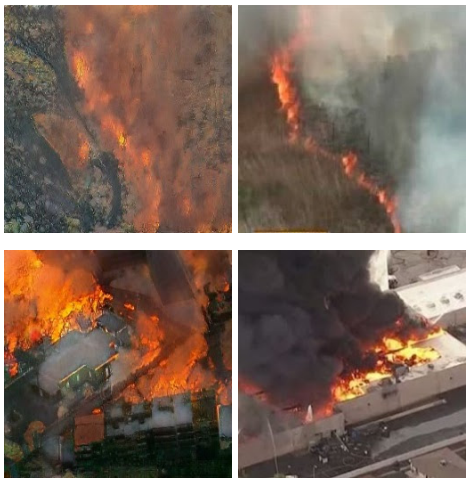

Figure 15: Two pairs of generated and real fire image samples, respectively. The top row depicts a forest scenario, while the bottom row depicts an urban scenario.

explained by a greater instability of the network for that same type of environment. Table 1 presents a summary of the statistics previously enumerated.

Table 1: Degree of realism of the generated fire images, evaluated on a scale of 1 to 5. Mdn, M and SD represent the median, the mean and the standard deviation values.

	Mdn	M	SD
Degree of realism (Overall)	4	3.44	1.159
Approximation (Forest)	4	3.52	0.893
Approximation (Urban)	3	2.98	1.064

We conclude that the images are of good quality and there is a particular preference for the ones generated for forest scenarios, which would be expected. On the other hand, the preference for bird's eye perspective over top-down perspective is not notorious. Interestingly, the respondents were only 3.6% more prone to select forestry image, of both top-down and bird's eye perspectives, than to select urban imagery. At the same time, they were also just 3% more confident that the images of top-down perspective were more realistic than the ones of bird's eye perspective. The difficulty respondents have had in making up their mind leads one to believe that, contrary to what one might have thought, the images from different scenarios and perspectives present a similar degree of quality and realism.

The generation anomalies identified concern mainly the lack of texture detail and distortions caused by exacerbated saturation levels or by the presence of artefacts such as noise. Also, some subjects expected a higher diversity of flames and smoke features and a higher image resolution, which they considered to have negatively impacted their assessment.

5.2 Fire Detection Performance

It was observed that the classification model maintains its original performance when facing the synthetic fire images. To test it, 876 aerial fire images were generated, both of urban and forest environments in the greater metropolitan areas of 4 cities, from Europe and California, in the United States.

The model reported an accuracy of 90.2% and a false positive rate of only 4.5%. Precision and recall tend to be inversely proportional to one another, that is, the increase of one usually implies the decrease of the other. This phenomenon occurred in this case, where for fire images the precision (94.9%) is higher than the recall (84,9%), trend that is reversed for non-fire images, where the precision (86.4%) is smaller than the recall (95.4%). The F1-score presents the harmonic mean between precision and recall and is useful to evaluate models when there is some imbalance in the class distribution. For the present case the F1-score is highly valued too, at 91.5%.

Another metric is the Area Under Curve (AUC), which measures the ability of the model to distinguish between the positive and the negative class based on the Receiver Operator Characteristic (ROC) curve, plotted using the true positive and false positive rates at various thresholds. The higher the value of AUC, comprised between 0 and 1, the better the classifier is able to distinguish between class samples and the better its predictive power. The current classifier is close to perfect at identifying the synthetic fire class, with an AUC of 94.8%.

Table 2 summarizes the results for the previously mentioned metrics.

Table 2: Classification metrics of the fire detector.

Class	Accuracy	Precision	Recall	F1-score	AUC
NoFire	0.902	0.864	0.954	0.907	0.948
Fire		0.949	0.849	0.896	
Mean	**0.902**	**0.916**	**0.915**	**0.915**	**0.948**

The confusion matrix of Fig. 16 shows, however, some discrepancy between the number of false negatives and false positives detected by the model. The first represent more than 15% of predictions, while the last only account for 4.5%.

After carefully analysing the constitution of each set, one comes to the conclusion that false positives are found to only contain images of forests, most of lower quality, as acquired from the external tiles provider, and being of bird's eye perspective. That may denote some overfitting of the model. Observe two false positive examples in Fig. 17.

On the other hand, false negative samples are mostly comprised of images of forestry with under-

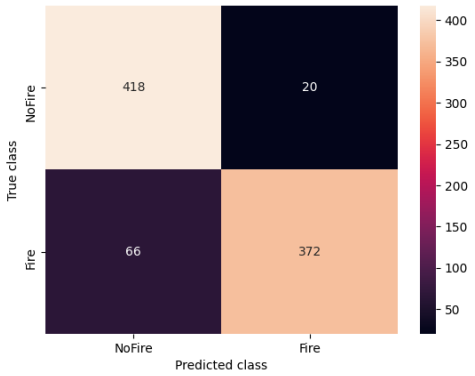

Figure 16: Confusion matrix of the fire detector on images of the validation dataset. False negatives and false positives account for 15% and 4.5% of predictions, respectively.

Figure 17: False positive fire samples.

growth and scattered vegetation, in which the generator tends to create undesired noise and blurring artifacts, vestigial columns of smoke but fails to generate flames. Some fire samples of urban scenarios suffering from color distortions are also wrongly classified by the model. Figure 18 displays one image sample representative of each case.

Figure 18: False negative fire samples.

The fact that the perspective of training images was variable brings some entropy to the ability of both the image generation and classification models. Fire detection in urban scenarios tends to portray worse results since the training of the generator was primarily focused on forest environments.

5.3 Performance Assessment

The processing time of the generation and detection pipeline is, at this stage, inherently dependent on the

generation process which, in turn, has a strong connection to the Bing Maps API. The servers' response time strongly affects the rate at which tiles can be provided to vehicles on a mission and, as a consequence, it has not been possible to simulate a camera as in a real-time scenario of 25 FPS. This would require a constant low latency connection to the Bing Maps servers and less restrictive measures to enable a larger number of requests for a given time interval.

We registered pipeline iterations of over 10 vehicle simulations using a camera of bird's eye perspective. With this configuration, the Camera Simulator requests two tiles to the Bing Maps API, for each vehicle's position. One is the original aerial tile, the other is similar but includes the fire polygon drawn on its appropriate location, should it exist on that image. The respective requests are summarized in the plots of Fig. 19. Note that the API was running without GPU in order to better approximate the behaviour of a machine with low computational resources.

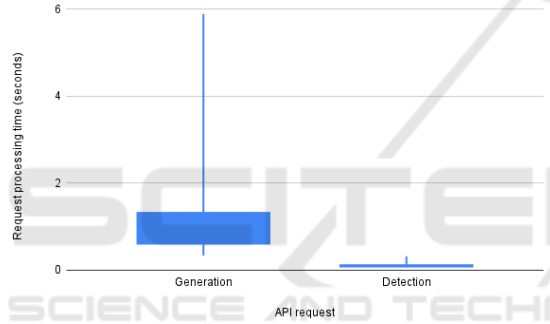

Figure 19: The processing time for the generation and detection requests reveal that the real-time bottleneck lies on the generation procedure.

In this plot we observe that the API is able to return at least an aerial tile every second for 50% of the collected request samples. The majority of these tiles do not contain features of fire, either because the vehicle's camera is not close enough to the burning area or because its orientation does not allow it to capture that region. This case, where the camera does not generate synthetic fire, constitutes the fastest response scenario for the vehicle and it still comprises about a second of tile fetching, rendering inadequate the realistic camera simulation and limiting the camera to a maximum of 1 FPS right from the start. In order to prevent overflowing the external Fire Module API with pending requests, especially when it comes to generating a sequence of fire tiles, the time interval between requests has been carefully set to 3 seconds.

The detection performance complies, on the other hand, with a scenario closer to real-time, averaging 100 ms per Vehicle Agent request at the same experiments, with as little as 30 ms of SD. Reducing the generation bottleneck would, according to these metrics, make it possible for the pipeline to run at around 10 FPS in the simulation environment, which is more acceptable. In reality, because a real aircraft would not need to simulate its own camera, the detector would be achieving over 50 FPS and consuming no more than 300 KB of memory. Therefore, it gathers all conditions necessary to run autonomously aboard an embedded system of low memory and storage resources (Kyrkou and Theocharides, 2019).

6 CONCLUSIONS AND FUTURE WORK

The models that are currently the reference in what concerns fire detection highlight a widespread problem which has been affecting their performance, the imbalance of classes in the training data, since there is a very small number of fire images, especially of an aerial perspective.

The generation of features of flames and smoke in images of aerial perspective is performed for the complete image, using a ResNet generator trained on image-to-image translation using the CycleGAN architecture. These are afterwards blended into the original image using the Cut-and-Paste technique in order to match them to the location of the burning region.

The similarity between generated and real images was assessed using the Fréchet Inception Distance. The declining trend of this metric, during training, denoted a gradual improvement of the generator, which produces images with increasingly less noise, blur and distortion. In addition, a group of 122 respondents to the conducted survey willingly provided their subjective opinion to evaluate the generated images qualitatively. These were considered of good quality, with a median realism of 4 out of 5, and proved to approximate images of forests better than those of urban environments, as initially intended and expected.

On the other hand, ERNet is a lightweight model designed to perform disaster detections in real-time, with good accuracy and low false positive rates on UAVs and similar CPU-based machines. It was adapted to perform binary classification on the existence of fire on the aerial images provided by the vehicles of *The Platform*. Not only did the detector manage to process requests in under 100 ms, but reached a high accuracy of 90.2% and confirmed the very low rates of false positives pledged by the original implementation, this time using generated images of fire. The false negatives accounted for 15.1% of cases and corresponded mainly to images of sparsely vegetated forest and urban scenarios with color distortions. The

false positives accounted for just 4.5% of predictions and contained only images of forest, most of them of low quality, which may evidence some overfitting of the detection model. Yet, given its AUC of 94.8%, we conclude that the model is able to identify very well the generated images of synthetic fire, further reinforcing the quality of the generator.

Integrated into the simulation platform, this module raises a number of questions, in particular concerning the generation procedure, because it is computationally more expensive than detection. An equilibrium was found to ensure its usability, but more can be done to improve it.

The implemented Fire Module interoperates with the external Bing Maps REST Services by means of HTTP requests. This communication may suffer from overheads, mostly because of the variable latency with the respective servers, which may also be overloaded and thus subject to longer response times. To tackle this problem it is essential to reduce the number of requests issued by creating caches to hold tiles of frequently used routes or by acquiring tiles of larger resolutions. The latter cover a larger surface area which can be segmented in order to match the onboard camera's field of view at the aircraft's position. Prefetching, a mechanism where tiles are retrieved in advance according to the predefined trajectory, could also prove beneficial.

The insertion of fire into bird's eye type of frames is currently subject to an internal functionality of the Bing Maps API which allows to perform the drawing of polygons on demand to be thereafter manually extracted. This implies that every drawing on the bird's eye perspective corresponds to an additional request to the external tiles provider, which is unfeasible. This issue should be resolved and considered for all other solutions that are subsequently integrated.

Since image generation proved to perform differently according to the environment, further developments could also separate the classification task into two specific models, training one of them on forestry while the other is trained on urban scenarios.

The incorporated lightweight detector based on ERNet portrays very promising results and opens up the opportunity to generalise the pipeline concept to other types of disturbances. It should help to identify, for example, building collapses, floods or traffic incidents already targeted by the detector. This would enable the comparison of different multi-vehicular approaches and help acquiring a deeper understanding on which works best for each case. One could therefore invest in studying the catastrophic scenarios from the air in order to define a sequence of priority actions to be carried out by the formation of aircraft.

REFERENCES

Almeida, J. (2017). Simulation and Management of Environmental Disturbances in Flight Simulator X. Master's thesis, University of Porto, Faculty of Engineering, Porto, Portugal.

Arcidiacono, C. (2018). An empirical study on synthetic image generation techniques for object detectors. Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), Stockholm, Sweden.

Borji, A. (2019). Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179:41–65. DOI:10.1016/j.cviu.2018.10.009.

Damasceno, R. (2020). Co-Simulation Architecture for Environmental Disturbances. Master's thesis, University of Porto, Faculty of Engineering, Porto, Portugal.

Dwibedi, D., Misra, I., and Hebert, M. (2017). Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1310–1319, Venice, Italy. IEEE Computer Society. DOI:10.1109/ICCV.2017.146.

El Harrouss, O., Almaadeed, N., Al-ma'adeed, S., and Akbari, Y. (2020). Image Inpainting: A Review. *Neural Processing Letters*, 51:2007–2028. DOI:10.1007/s11063-019-10163-0.

Filippi, J. B., Bosseur, F., and Grandi, D. (2014). *ForeFire: open-source code for wildland fire spread models*, pages 275–282. Advances in Forest Fire Research. Imprensa da Universidade de Coimbra, Coimbra, Portugal. DOI:10.14195/978-989-26-0884-6_29.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Proceedings of The 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Montreal, Canada. MIT Press. DOI:10.1145/3422622.

Hartin, E. (2008). Fire Development and Fire Behavior Indicators. Technical report, Compartment Fire Behavior Training.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA. IEEE Computer Society. DOI:10.1109/CVPR.2016.90.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Long Beach, California, USA. Curran Associates Inc.

Hollosi, J. and Ballagi, A. (2019). Training Neural Networks with Computer Generated Images. In *Proceedings of 2019 IEEE 15th International Scientific Conference on Informatics*, pages 155–160, Poprad, Slovakia. IEEE Computer Society. DOI:10.1109/Informatics47936.2019.9119273.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, Honolulu, Hawaii, USA. IEEE Computer Society. DOI:10.1109/CVPR.2017.243.

Huang, X. and Belongie, S. (2017). Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, Venice, Italy. IEEE Computer Society. DOI:10.1109/ICCV.2017.167.

Kyrkou, C. and Theocharides, T. (2019). Deep-Learning-Based Aerial Image Classification for Emergency Response Applications Using Unmanned Aerial Vehicles. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 517–525, Long Beach, California, USA. IEEE Computer Society. DOI:10.1109/CVPRW.2019.00077.

Liu, H., Jiang, B., Xiao, Y., and Yang, C. (2019). Coherent Semantic Attention for Image Inpainting. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4169–4178, Seoul, Korea. IEEE Computer Society. DOI:10.1109/ICCV.2019.00427.

Microsoft (2021). Bing Maps Licensing. Online: https://www.microsoft.com/en-us/maps/licensing. (accessed 2021-02-11).

Mograph Plus (2018). Corona for 3ds Max — Rendering Smoke, Fire and Explosions — Tutorial #113. Online: https://www.youtube.com/watch?v=DYTDNGqvPUw. (accessed 2021-11-23).

Park, M., Tran, D. Q., Jung, D., and Park, S. (2020). Wildfire-Detection Method Using DenseNet and CycleGAN Data Augmentation-Based Remote Camera Imagery. *Remote Sensing*, 12(22):3715. DOI:10.3390/rs12223715.

Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson Image Editing. *ACM Transactions on Graphics - TOG*, 22(3):313–318. DOI:10.1145/882262.882269.

PORDATA (2020). Forest fires and burn area. Online: https://www.pordata.pt/Europa/Incłndios+florestais+e+rea+ardida-1374. (accessed 2021-01-27).

Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Transactions on Graphics - TOG*, 23(3):309–314. DOI:10.1145/1015706.1015720.

Silva, D. C. (2011). *Cooperative multi-robot missions : development of a platform and a specification language*. PhD thesis, University of Porto, Faculty of Engineering, Porto, Portugal.

Silva, T. (2017). A Short Introduction to Generative Adversarial Networks. Online: https://sthalles.github.io/intro-to-gans/. (accessed 2020-12-16).

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of The 3rd International Conference on Learning Representations (ICLR)*, San Diego, California, USA. arXiv: 1409.1556 [cs.CV].

Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J. M., and Chari, V. (2019). Learning to Generate Synthetic Data via Compositing. In *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 461–470, Long Beach, California, USA. IEEE Computer Society. DOI:10.1109/CVPR.2019.00055.

Turco, M., Jerez, S., Augusto, S., Tarín-Carrasco, P., Ratola, N., Jiménez-Guerrero, P., and Trigo, R. M. (2019). Climate drivers of the 2017 devastating fires in Portugal. *Scientific Reports*, 9:13886. DOI: 10.1038/s41598-019-50281-2.

WMO (2021). WMO Atlas of Mortality and Economic Losses from Weather, Climate and Water Extremes (1970–2019). Technical Report WMO- No. 1267, World Meteorological Organization. ISBN: 978-92-63-11267-5.

Wong, M. Z., Kunii, K., Baylis, M., Ong, W. H., Kroupa, P., and Koller, S. (2019). Synthetic dataset generation for object-to-model deep learning in industrial applications. *PeerJ Computer Science*, 5:e222. DOI:10.7717/peerj-cs.222.

Zhu, J., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Venice, Italy. IEEE Computer Society. DOI:10.1109/ICCV.2017.244.