mathematics

MDPI

Article

Reducing the Reality Gap Using Hybrid Data for Real-Time Autonomous Operations

Suleyman Yildirim [1,*] and Zeeshan A. Rana [2,*]

[1] Digital Aviation Research and Technology Centre (DARTeC), Cranfield University, Bedford MK43 0AL, UK
[2] Centre for Aeronautics, Cranfield University, Bedford MK43 0AL, UK
* Correspondence: suleyman.yildirim@cranfield.ac.uk (S.Y.); zeeshan.rana@cranfield.ac.uk (Z.A.R.)

Abstract: This paper presents an ablation study aimed at investigating the impact of a hybrid dataset, domain randomisation, and custom-designed neural network architecture on the performance of object localisation. In this regard, real images were gathered from the Boeing 737-400 aircraft while synthetic images were generated using the domain randomisation technique involved randomising various parameters of the simulation environment in a photo-realistic manner. The study results indicated that the use of the hybrid dataset, domain randomisation, and the custom-designed neural network architecture yielded a significant enhancement in object localisation performance. Furthermore, the study demonstrated that domain randomisation facilitated the reduction of the reality gap between the real-world and simulation environments, leading to a better generalisation of the neural network architecture on real-world data. Additionally, the ablation study delved into the impact of each randomisation parameter on the neural network architecture's performance. The insights gleaned from this investigation shed light on the importance of each constituent component of the proposed methodology and how they interact to enhance object localisation performance. The study affirms that deploying a hybrid dataset, domain randomisation, and custom-designed neural network architecture is an effective approach to training deep neural networks for object localisation tasks. The findings of this study can be applied to a wide range of computer vision applications, particularly in scenarios where collecting large amounts of labelled real-world data is challenging. The study employed a custom-designed neural network architecture that achieved 99.19% accuracy, 98.26% precision, 99.58% recall, and 97.92% mAP@.95 trained using a hybrid dataset comprising synthetic and real images.

Keywords: hybrid data; synthetic data; object detection; ablation study

MSC: 68T40

Citation: Yildirim, S.; Rana, Z.A. Reducing the Reality Gap Using Hybrid Data for Real-Time Autonomous Operations. *Mathematics* **2023**, *11*, 1696. https://doi.org/10.3390/math11071696

Academic Editors: Andrey Gorshenin, Mikhail Posypkin and Vladimir Titarev

Received: 13 February 2023
Revised: 17 March 2023
Accepted: 29 March 2023
Published: 2 April 2023

1. Introduction

Collection and manual annotation of a large amount of data is an expensive and time-consuming task, yet it is a highly critical stage for training and testing a neural network. The requirement of this stage becomes more problematic when the collection of images in large numbers and variations is needed, the labels are hard to specify manually, or the task depends on expert knowledge. Pixel-perfect segmentation or three-dimensional poses takes a significant amount of time for a human being to manually annotate a single image. A favourable method—three-dimensional computer graphics software tools, such as Blender [1], Unreal Engine [2], and Unity [3]—has been adopted to generate the pixel perfectly and automatically annotated synthetic data. Some of the studies [4–15] published in recent years have been using generated datasets in a simulation environment. To generate these kinds of datasets, the models need to be carefully designed in significant detail. The datasets above have been used to train neural networks for pose estimation, optical flow, scene flow, and stereo disparity estimation problems.

Although the large quantity of annotated data is freely available for object detection, sophisticated object detection problems could benefit from photo-realistic synthetic image data generation to reduce the number of required ground truth data. Without abandoning photo-realism by randomly altering the synthetic environment in different ways, a domain randomisation solution [16] has been proposed to lead the neural network to focus on the rudimentary attributes of the object. This method has been applied successfully on various tasks such as three-dimensional coordinate detection of coloured cubes on the table [16] and determination of control commands of a quad-copter indoor environment [17], as well as scene flow [8] and optical flow [5].

In this study, the domain randomisation method has been extended by using hybrid image data and a custom-designed neural network to locate the pressure refuelling adaptor in three-dimensional space in real time. This research has been conducted to seek answers to the questions below:

- Can synthetic image data generation achieve effective results on real-world problems with the help of domain randomisation?
- Which parameters of domain randomisation affect the results most?
- How much does the augmentation improve the accuracy of the neural network?
- Which layers of the neural network affect the accuracy most?
- What are the main benefits of using hybrid image data to train the neural network?
- Can the synthetic image data be fully relied on to train the neural network?

As the introduction to the ablation study is presented in Section 1, the rest of the paper is organised as follows: Section 2 outlines existing methods related to the ablation study. A brief explanation of the proposed method can be found in Section 3. The results and discussion can be found in Sections 5 and 6, respectively. Finally, the conclusion is presented in Section 7.

2. Literature Review

In recent years, the popularity of synthetic data used for testing and training purposes has risen. A large number of datasets—Virtual KITTI [6], SceneNet RGBD [9], Flying Chairs [5], SYNTHIA [13], UnrealStereo [11,15], FlyingThings3D [8], SceneNet [7], MPI Sintel [4], Sim4CV [10], GTA V [12]—could be given as an example. To solve computer vision problems, such as stereo disparity estimation, scene flow, camera pose estimation, and optical flow, the datasets above have been generated.

Even though some of the datasets have solely been trained on synthetic data, these datasets contain both semantic segmentation masks and object detection annotations. By adding Gaussian blurring to the object's edges and Gaussian noise to the object, Hinterstoisser et al. [18] used this synthetic dataset to train the final layers of their neural network while the rest of the network was pre-trained with real data only. It has been observed that their approach did not increase the success of the neural network. On the contrary, training the final layers of the neural network with only synthetic data was rather harmful.

Tobin et al. [16] proposed to close the reality gap by using domain randomisation as an alternative method to high-quality synthetic data. They generated a synthetic image dataset in different variations so the neural network sees the data as another version of the real world. Domain randomisation has been used in their research to train a neural network to locate the different shape-based objects in three-dimensional space to manipulate the robotic arm.

Introduction to the domain randomisation method was conducted by Sadeghi and Levine [17], who used synthetic images to train a quadcopter to fly in indoor environments. The FlyingThings3D [8] and Flying Chairs [5] datasets can be considered different variations of the domain randomisation method.

Domain randomisation has also been applied to robotics control policy. While James et al. [19] used domain randomisation to make a robot pick up a cube and place it in a basket, Zhang et al. [20] used the method to manipulate the robot near a cube. Other studies have adopted the domain randomisation method to explore robotic policies from a

high-quality rendering engine [21], train an object classifier from three-dimensional CAD models [22], and generate a high-quality synthetic image dataset [14].

Dwibedi et al. [23] proposed a similar approach by adding object images onto the background images. The accurate object segmentation with this method was highly time-consuming and challenging. These two problems with their methods outline the drawback of this approach.

3. Methodology

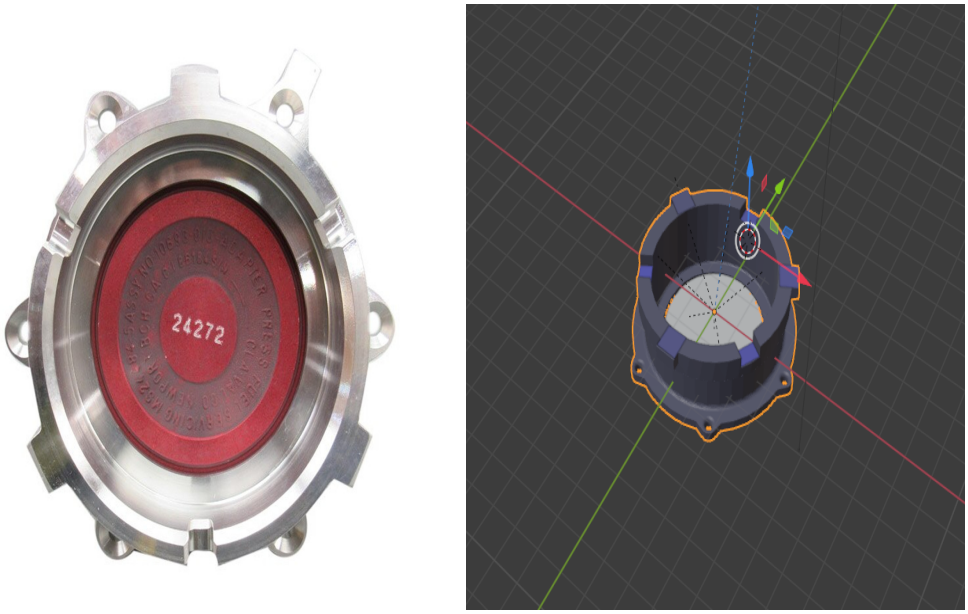
3.1. Dataset Development

The pressurised fuel adaptor, also known as a bottom loading adaptor [24], is a device that connects to an aircraft and supplies it with pressurised fuel, as illustrated in Figure 1a. The pressurised refuelling adaptor must be designed and built to meet both MS24484-5 and MIL-A-25896 standards. "MIL-STD" [25], which stands for Military Standard, is a set of standards established by the United States Department of Defence to ensure standardisation in its operations. It is used to meet the standardisation objectives of the Department of Defence. Standardisation is useful in achieving a variety of goals, including commonality, interoperability, reliability, cost efficiency, compatibility with logistics systems, and compliance with defence-related requirements. It helps to ensure that products meet specific requirements [26]. The pressurised refuelling adaptor is made of high-strength stainless steel and aluminium to ensure maximum strength and durability.

The quality of the dataset is important, so it is necessary to remove any insufficient data that does not have a use. The term 'quality' is subjective and can be difficult to define precisely. In a broad sense, quality can be understood as the approach or option that provides the best results when evaluated empirically. Having a high-quality dataset is crucial for effectively tackling problems and finding the optimal solution. A dataset that is accurate, relevant, and complete can provide valuable and reliable information that can be used to make informed decisions or identify the most effective solution to a problem. Defining what constitutes high-quality data is important during the process of creating a dataset. Having a clear understanding of the characteristics of good data can help ensure that the dataset being developed is accurate, relevant, and complete, which are all important factors in making the dataset useful and effective. A high-quality dataset has specific characteristics that contribute to better performance in terms of feature representation, reducing skew, and increasing reliability [27].

Articulating the problem clearly is the most important step in the dataset development process before generating or collecting high-quality data. Determining how to collect the necessary data, what data to collect, and what to predict with that data are crucial steps in clearly defining the problem. Without a clear understanding of these factors, it may be difficult to develop a dataset that is accurate, relevant, and complete. Before developing a solution or exploring the data, it is important to identify the category of the problem being addressed, which could be clustering, regression, or classification. This will help guide the approach to formulate a solution and analyse the data [28].

Manually collecting data can be a tedious and burdensome task, which is why it is essential to establish effective data collection mechanisms during the preparation of a dataset. Automating data collection can help eliminate the need for repetitive manual data collection, saving time and reducing the risk of errors. The dataset collection process has been divided into two stages due to the limitations of both approaches. The first stage of the dataset collection process involves setting up a camera rig system to collect real data from a Boeing 737-400.



(**a**) (**b**)

Figure 1. Main Elements of Hybrid Dataset. (**a**) Pressurised Refuelling Adaptor [29]. (**b**) 3D Design of Refuelling Adaptor.

The steps below have been followed to set up the camera rig system to collect the real dataset.

- A sturdy mounting system: To capture clear and stable images of the refuelling adaptor, it is crucial to have a sturdy mounting system that can securely attach to the Intel® RealSense™ D435 depth camera. The system should be designed to minimise any movement or vibrations, as this can result in blurry or distorted images. Additionally, the mounting system should be adjustable to ensure the camera is positioned at an optimal distance and angle to capture the refuelling adaptor and surrounding area.
- High-resolution cameras: The camera used for this application should be of high resolution to capture clear and detailed images of the refuelling adaptor. The resolution should be high enough to capture fine details.
- Appropriate distance and angle: The Intel® RealSense™ D435 depth camera has been positioned at an appropriate distance and angle to capture the refuelling adaptor and surrounding area to capture the images. The ideal position will depend on the specific aircraft and refuelling setup and may require some experimentation to find the optimal position. However, the Intel® RealSense™ D435 depth camera should be positioned to capture a wide field of view that includes the entire refuelling adaptor and any relevant surrounding components.
- Consistent lighting conditions: To ensure consistent lighting conditions across all captured images, an array of high-intensity LED lights could be positioned around the camera rig. The LED lights should be positioned and angled to provide even illumination of the refuelling adaptor without creating harsh shadows or over-exposed areas. This is important to ensure that the images are clear and easy to interpret and that any potential issues or anomalies during the process are clearly visible.

The synthetic dataset was produced by designing a 3D model of the object and adding different materials to it. The model was modified to simulate different weather and lighting conditions, allowing for the creation of a synthetic dataset [28].

To generate a photo-realistic synthetic representation of the refuelling adaptor and its environment, several steps need to be taken.

- Model the refuelling adaptor: This has been accomplished by importing the 3D model of the adaptor. It is important to ensure that the model is accurate and to scale to ensure a realistic final product.
- Texture the models: To make the model looks more realistic, textures and materials need to be added to them. This involves adding textures and materials to the model in Blender. It is important to consider factors such as the paint and the material of the refuelling adaptor.
- Set up lighting and virtual camera: The lighting and virtual camera setup are crucial components in creating realistic images. Appropriate lighting needs to be set up in the 3D environment to create shadows and reflections that are similar to those found in real life. This involves adding lights to the scene in Blender and using ZPy to programmatically change the lighting to cover every potential scenario. A virtual camera also needs to be set up to capture the images. This was done by positioning a virtual camera in Blender and ZPy to program the camera position.
- Render the images: The final step in the process is to render the images. This has been done using Blender's built-in rendering engine, which offers a variety of settings to create high-quality images. It is important to consider factors such as the resolution of the images to ensure the highest quality output.

In Figure 1a,b the elements used in the development of both real and synthetic datasets are shown.

The hybrid dataset used for training and testing consists of 770 training images, 74 validation images, and 37 test images [30]. These images depict the refuelling adaptor of an Airbus 737-400 aircraft. The input image specifications and pre-processing steps play a crucial role in the performance of the image classification model. In this case, the input images were annotated in COCO format, which is a widely-used format for object detection tasks. Before feeding the images to the model, the pre-processing steps were applied to ensure that the model receives the most relevant and useful information from the images. These pre-processing steps included applying a 50% probability of horizontal and vertical flips, as well as rotating the image by 90 degrees in clockwise, counter-clockwise, and upside-down directions with equal probability. This helps the model learn from a diverse set of images and angles, which can be useful in handling real-world scenarios. Additionally, a random Gaussian blur with a radius between 0 and 1 pixels was applied to each image, which can help to reduce noise and improve the overall quality of the image. These pre-processing steps help to ensure that the model receives a consistent and high-quality input that can improve the accuracy and robustness of the model [31]. The process of creating a synthetic dataset involves several steps. Once the 3D refuelling adaptor model is available, textures representing materials such as high-strength stainless steel and aluminium are applied to them. To create a more realistic environment, HDRI maps were used to simulate realistic lighting conditions. These maps capture a wide range of light intensities in a scene allowing for more accurate lighting in synthetic images. In addition, weather conditions such as rain, snow, and fog were added to simulate real-world scenarios. To provide variations, the synthetic dataset images were rendered under different lighting conditions. The refuelling adaptor was captured under bright sunlight and overcast skies with artificial lighting. Different camera positions, orientations, and fields of view were also used to generate variations in the images.

As shown in Figure 2, the hybrid dataset includes sample images of the refuelling adaptor. To further enhance the dataset, synthetic images were generated using a 3D CAD model. These synthetic images incorporate a variety of textures, HDRIs (High Dynamic Range Images), weather conditions, and lighting scenarios, which helps to diversify and expand the range of the dataset. Figure 3 showcases some examples of these generated images.

(**a**) Sample 1 (**b**) Sample 2 (**c**) Sample 3

Figure 2. Sample Images from Real Dataset.

(**a**) Sample 1 (**b**) Sample 2 (**c**) Sample 3

Figure 3. Sample Images from Synthetic Dataset.

3.2. Domain Randomisation

Domain randomisation is a technique that involves generating hundreds of variations of an object and its environment to make it easier for a machine learning model to identify patterns. One issue that often arises when using synthetic data is the domain gap, which refers to the difference in prediction accuracy between a model trained on synthetic data and a model trained on real data. Domain randomisation can help minimise this gap by improving the accuracy of the machine learning model on synthetic data [32]. This makes an essential element of the synthetic dataset generation process in this research.

The limitations of both the real and synthetic datasets are as follows:

Limitations of the real dataset can be listed as:

- The process of collecting and annotating thousands of images can be time-consuming;
- Even though there are many freely available datasets, the dataset needs to be collected and annotated for custom objects;
- Annotations are generally created by humans and humans tend to make mistakes;
- The content of the dataset might involve the wrong classes of images;
- Real datasets may only include basic annotations such as bounding boxes, segmentation, or labels.

Limitations of the synthetic dataset can be listed as:

- While synthetic data can replicate many of the properties of real data, it may not be able to accurately replicate all aspects of the original content, which can negatively impact the accuracy of the model.
- The quality of the generated data is heavily dependent on the quality of the 3D model.

The goal of this research is to address these limitations by combining these two datasets in order to take advantage of the benefits of both techniques.

3.3. Tranining Neural Networks

The custom neural network was designed based on three principles to achieve high accuracy while minimising computational cost and maintaining high fps in real-time operation. These principles are "Compound Scaling, Neural Architecture Search, and Inverted Residual Block". These techniques have been used to improve the accuracy of various neural networks. To maximise the benefits of these techniques, they were combined to create a custom neural network. The following section explains how these techniques contributed to the improved accuracy of the network.

Prior to the introduction of compound scaling with EfficientNet, the most prevalent method of scaling neural networks was to increase either their dimensions, i.e., height, width, depth, or image size [33]. EfficientNet's compound scaling approach scales the depth, width, and resolution of the network uniformly using a set of fixed scaling coefficients, in

contrast to the conventional practice of arbitrarily scaling these factors. To utilise 2^N times more computation power, the constants α, β, and γ were determined through grid search on the original model and used to increase the depth by α^N, the width by β^N, and the image size by γ^N. Rather than utilising different coefficients for each dimension, EfficientNet employs a compound coefficient ϕ to uniformly scale the network. As the convolutional neural network requires additional layers to capture fine-grained patterns as the input image increases in size, balancing all dimensions of the network leads to better overall performance compared to scaling depth, width, and resolution using different coefficients.

Neural architecture search is a method based on reinforcement learning that involves developing a baseline neural architecture using a multi-objective search that optimises accuracy and FLOPS (floating-point operations per second) as the optimisation goals, rather than latency. The objective function serves as a control mechanism to identify the highest-performing model in terms of accuracy and FLOPS. The controller defines the model architecture, which is then used for training. After each training sequence, a reward function calculates and provides feedback to the controller to define a new model architecture. This process repeats until the best-performing architecture is identified based on the given accuracy and latency goals. In MNasNet [34], the objective function is defined as $ACC(m) \times [FLOPS(m)/T]^w$.

Inverted residual blocks, depicted in Figure 4 below, are implemented in MBConv and were first introduced in Inception-V2 [35]. Rather than decreasing the number of channels, the inverted residual block increases it by a factor of three. As standard convolution operations are computationally intensive, a depthwise convolution operation is utilised to generate the output feature map. The second convolution layer reduces the number of channels in the final stage.

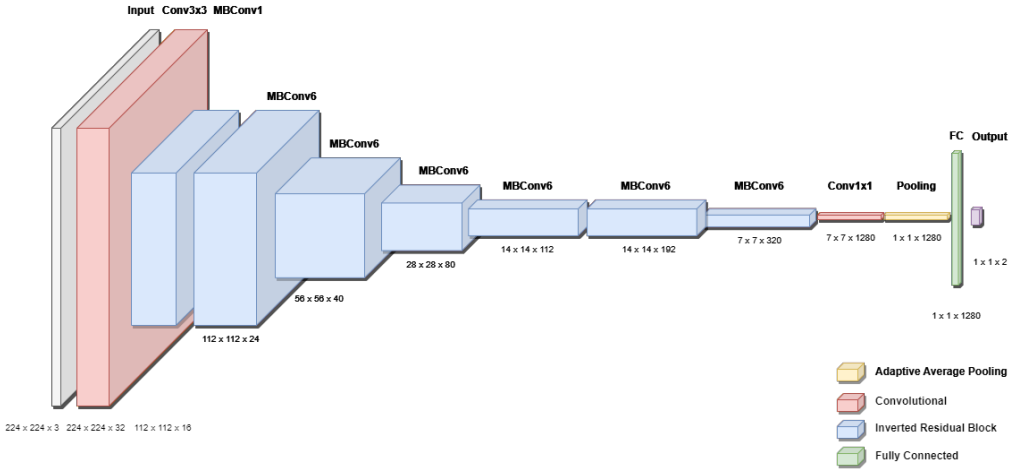


Figure 4. Custom Neural Network.

The hyperparameters for the custom-designed neural network are as follows: The Width Multiplier ϕ has been implemented to scale the number of channels in each layer, thereby controlling the size of the network, and its default value is 1.2. The Depth Multiplier α was used to control the depth of the network by scaling the number of layers in the network with a default value of 1.2. The Resolution Multiplier ρ was used to scale the size of the input image, thereby controlling the input resolution of the network; its default value was set to 1.1. To prevent over-fitting, a Dropout Rate of 0.25 was implemented, which drops a unit in the network during training. The network was configured with 16 blocks, each containing a series of layers with a specific set of hyperparameters such as the number of filters, kernel size, etc. The Expansion Factor γ was used to control the expansion of the network by scaling the number of channels in the first convolutional layer of each block with a default value of 1.0. The Convolutional Kernel Size and the Depthwise Convolutional Kernel Size were set to 5×5 for each layer and the depthwise convolutional

layer of the network. The Stride used in the convolutional and depthwise convolutional layers of the network was 1.

3.4. Ablation Study

Ablation studies can be used to evaluate the importance of different components in neural network training for object detection using real and synthetic datasets. By ablating one or more components from the neural network and comparing the performance of the modified neural network to the original neural network, the contribution of each element to the overall performance of the neural network can be identified. The ablation study evaluates the importance of different features or feature extractors in neural network training for object detection. Ablation studies can also be used to evaluate the importance of different types of layers in neural networks, such as convolutional layers, pooling layers, and fully connected layers. The performance of a neural network is compared with only hand-crafted features to one with a combination of hand-crafted and learned features, or a pre-trained convolutional neural network as a feature extractor to one custom-designed neural network. By comparing the performance of these different configurations and the performance of a neural network with only certain types of layers to one with a combination of layers on real and synthetic datasets, the ablation study helps us identify which type of features or feature extractors and which types of layers are most effective for the refuelling adaptor detection task [36].

Overall, ablation studies can provide valuable insights into the strengths and weaknesses of a CNN or custom-designed neural network model for object detection, and the opportunities for improvement can be identified. By understanding the contribution of different components to the performance of the neural network on both real and synthetic datasets, a more effective and efficient neural network model can be designed for the refuelling adaptor detection task.

3.5. Experimental Setup

The real dataset for this study was obtained from Cranfield University's Boeing 737-400 aircraft. To supplement the real data, a synthetic dataset was generated using the Blender and zpy [37] open-source computer vision toolkit. The custom-designed neural network and transfer learning models were trained on the HILDA high-performance computer at Cranfield University, which was equipped with 112 Intel Xeon Gold 6258R CPU cores, 4 NVIDIA A100 80GB GPUs, 377GB of DDR4-2933 RAM, and 330Tb of storage capacity. These resources were allocated specifically for this research project.

4. Results

In this section, a thorough explanation of the parameters that are utilised to design the custom neural network model is made. The Comparison Table 1 of the custom-designed neural network model with the pre-trained models, e.g., EfficientNet-B0, VGG-16, and ResNet-18, is established in terms of data type, learning rate, optimiser function, accuracy, validation loss, precision, recall, and $mAP@.95$.

Compound Scaling is a method for scaling up the size and capacity of a neural network that has been shown to be effective in improving the accuracy of the neural network model for the refuelling adaptor detection task. It involves increasing the number of channels in the convolutional layers and the number of layers in the network, while also reducing the spatial resolution of the intermediate feature maps.

One of the main benefits of Compound Scaling for object detection is that it allows the neural network to extract and process more discriminating, high-level features from the input data. In object detection, the goal is to identify and classify objects in images or video, and the accuracy of the neural network model depends heavily on its ability to extract features that are relevant to the refuelling adaptor detection task. By increasing the size of the neural network and reducing the spatial resolution, the neural network model is able to capture more abstract, semantically meaningful features that are more indicative of the presence and type of objects in the scene.

Table 1. Ablation Study Results.

Neural Network Techniques	Data Type	Learning Rate	Optimiser Function	Accuracy	Validation Loss	Precision (%)	Recall (%)	*mAP*@.95 (%)
Compound Scaling Neural Architecture Search Inverted Residual Block	Real Dataset	10^{-3}	Adam	46.36	23.981	38.30	37.63	41.48
			SGD	47.22	22.585	39.31	38.46	28.49
			Rmsprop	42.87	26.458	36.12	35.22	29.21
Compound Scaling Neural Architecture Search Inverted Residual Block	Hybrid Dataset	10^{-3}	Adam	52.19	19.329	41.27	46.38	38.58
			SGD	56.44	15.832	43.83	48.37	40.37
			Rmsprop	47.67	20.832	42.14	42.57	39.47
Compound Scaling Neural Architecture Search Inverted Residual Block	Real Dataset	10^{-4}	Adam	72.8	14.254	61.74	62.19	53.85
			SGD	77.42	12.832	62.18	63.28	54.32
			Rmsprop	64.13	16.239	60.49	60.43	51.32
Compound Scaling Neural Architecture Search Inverted Residual Block	Hybrid Dataset	10^{-4}	Adam	73.5	10.329	63.21	64.20	57.47
			SGD	78.43	9.848	64.37	65.33	58.38
			Rmsprop	68.79	12.328	62.47	63.27	56.47
Compound Scaling Neural Architecture Search Inverted Residual Block	Real Dataset	10^{-5}	Adam	91.87	0.325	92.22	94.67	92.57
			SGD	94.31	0.209	93.83	95.88	93.24
			Rmsprop	86.99	0.465	90.62	94.19	91.42
Compound Scaling Neural Architecture Search Inverted Residual Block	Hybrid Dataset	10^{-5}	Adam	98.37	0.044	97.03	98.34	97.14
			SGD	99.19	0.023	98.26	99.58	97.92
			Rmsprop	97.56	0.078	95.47	96.01	96.67
EfficientNet-B0	Real Dataset	10^{-5}	Adam	72.82	6.449	61.40	60.14	56.16
			SGD	76.11	6.214	62.89	60.98	55.37
			Rmsprop	75.37	8.823	60.95	59.16	54.56
EfficientNet-B0	Hybrid Dataset	10^{-5}	Adam	78.09	5.382	64.31	65.35	58.43
			SGD	83.11	4.974	66.49	67.15	59.47
			Rmsprop	82.47	6.238	62.58	64.75	56.48
VGG-16	Hybrid Dataset	10^{-5}	Adam	80.88	2.374	84.74	83.19	82.48
			SGD	85.56	2.249	86.38	86.17	83.29
			Rmsprop	81.74	3.958	81.36	80.12	80.44
ResNet-18	Hybrid Dataset	10^{-5}	Adam	87.44	0.402	90.11	92.89	91.23
			SGD	90.89	0.388	92.18	93.77	91.88
			Rmsprop	88.32	0.627	88.76	91.03	90.58

In addition, Compound Scaling helps reduce the over-fitting of the neural network model to the training data. As the neural network becomes larger and more expressive computationally, it is able to capture more of the underlying structure of the data.

Neural Architecture Search is a technique that automates the process of designing and optimising the architecture of the neural network. It has been shown to be effective in improving the accuracy of the neural network model for the refuelling adaptor detection task. As the network must be able to extract and process features that are relevant to the refuelling adaptor detection task, traditional hand-designed architectures can be time-consuming and labour-intensive to design and optimise, and may not always result in the best performance.

NAS algorithms search through a large space of possible network architectures to find the one that performs the best on the refuelling adaptor detection task at hand. NAS algorithms do this by using a search algorithm, such as evolutionary search or reinforcement learning, to iteratively explore the space of possible architectures and select the ones that perform the best. The search process can be guided by a performance metric, such as accuracy or mAP, and the resulting architecture can be fine-tuned using traditional hyperparameter optimisation techniques.

By using NAS to find an optimal network architecture, the accuracy of the neural network model can be improved. NAS has been shown to be effective in finding architectures that outperform hand-designed ones, and it helps reduce the human effort required to design and optimise the architecture of a neural network.

Inverted Residual Blocks are a type of building block that can be used in CNNs and have been shown to be effective in improving the accuracy of CNNs for the refuelling adaptor detection task. Inverted Residual Blocks are designed to improve the efficiency of the network by using point-wise convolutions to reduce the number of channels in the intermediate feature maps, rather than using traditional convolutions to increase the number of channels. This improves the accuracy of the neural network model by allowing it to process more information with fewer parameters, which reduces the risk of over-fitting and improves the generalisation performance of the neural network model. Inverted Residual Blocks also use a shortcut connection that allows the neural network model to skip layers and directly access deeper features. This helps improve the accuracy of the neural network model by allowing it to directly access and process semantically meaningful features that are more indicative of the presence and type of objects in the scene.

The custom-designed neural network stands out with its results in Table 1 as it employs the methods discussed in the above paragraphs. Here are a few reasons why the custom-designed neural network is better than pre-trained models, such as EfficientNet-B0, VGG-16, and ResNet-18.

- Task-specific design: A custom-designed neural network is specifically designed and optimised for the refuelling adaptor detection task, while pre-trained models are usually designed to be versatile and adaptable to a wide range of tasks. As a result, a custom-designed network is able to outperform pre-trained models on the refuelling adaptor detection task.
- Hyper-parameter optimisation: Pre-trained models are generally trained using a fixed set of hyperparameters, whereas the custom-designed neural network is fine-tuned using techniques such as hyper-parameter optimisation to find the best set of hyperparameters for the refuelling adaptor detection task. This helps improve the performance of the custom-designed network.
- Dataset characteristics: Pre-trained models are trained on large datasets that may have different attributes from the dataset that has been used for this research. The custom-designed neural network is trained specifically on the hybrid dataset, which resulted in better performance.
- Architectural differences: Pre-trained models have a fixed architecture that may not be optimal for every task. The custom-designed neural network is designed with

an architecture that is more suitable for the refuelling adaptor detection task, which resulted in improved performance.

The hybrid dataset plays a critical role in obtaining these results. Here are a few reasons why a hybrid dataset is advantageous in this research:

- Increased diversity: Hybrid dataset offers a greater variety of data compared to solely real or synthetic datasets. This is especially useful as the refuelling adaptor detection task requires the neural network model to generalise to a wide range of conditions and the real dataset is limited in size or diversity.
- Improved annotation quality: Synthetic dataset has precise, accurate annotations, while the real dataset may have less accurate or incomplete annotations. By combining real and synthetic data, it is possible to take advantage of the precise annotations in the synthetic data while also incorporating the complexity and variability of real data.
- Reduced cost and ethical concerns: Synthetic dataset can be generated at a lower cost and with fewer ethical concerns than a real dataset. By using a hybrid approach, it is possible to reduce the amount of real data that needs to be collected, while still incorporating the benefits of real data.

Table 1 presents the results of an ablation study conducted on two datasets: Real and Hybrid. The table is divided into six sections based on the different optimiser functions Adam, SGD, and Rmsprop. For each optimiser function, three experiments were conducted with different learning rates $1e^{-3}$, $1e^{-4}$, and $1e^{-5}$. The table shows the accuracy, validation loss, precision, recall, and mAP@.95% of each experiment. The results show that the hybrid dataset performs better than the real dataset for all optimiser functions and learning rates with the highest accuracy achieved using SGD optimiser with a learning rate of $1e^{-5}$, reaching 99.19% accuracy, 0.023 validation loss, 98.26% precision, 99.58% recall, and 97.92% mAP@.95%. The detailed results can be seen below.

In this study, we proposed a custom neural network that utilises compound scaling, neural architecture search (NAS), and inverted residual blocks, and trained it using the Stochastic Gradient Descent (SGD) optimiser function. We evaluated our model on a hybrid dataset. The results showed that our model achieved an outstanding accuracy of 99.19%. Our model also showed high precision of 98.26%, recall of 99.58% and mAP@.95 of 97.92%. These results indicate that our custom neural network was able to effectively learn the features of the refuelling adaptor and make accurate predictions. Our model performed better than the baseline models, such as EfficientNet-B0, VGG-16, and ResNet-18, in terms of accuracy, precision, recall, and mAP@.95. These results demonstrate the effectiveness of the proposed architecture in refuelling adaptor detection in real-time, and the potential of compound scaling, NAS, and inverted residual blocks in improving the performance and efficiency of neural networks. The use of the SGD optimiser function helped the model converge quickly and efficiently to the optimal solution, resulting in a highly accurate model.

In Figure 5, it is evident that the custom-designed neural network is capable of detecting the refuelling adaptor in real-time on a Boeing 737-400 aircraft. This is a significant achievement as it highlights the potential of utilising advanced machine learning techniques for real-world applications. Furthermore, the ability to detect the refuelling adaptor in real time suggests that this custom-designed neural network has the capability to process and analyse data in a timely manner which is crucial for efficient and effective decision-making. Overall, this research provides compelling evidence of the effectiveness of utilising a hybrid dataset for real-time detection tasks. The use of a combination of different data sources, such as both synthetic and real-world data, has been shown to improve the performance of the detection model. This is because it allows the model to learn from a diverse range of examples and generalise better to new situations.

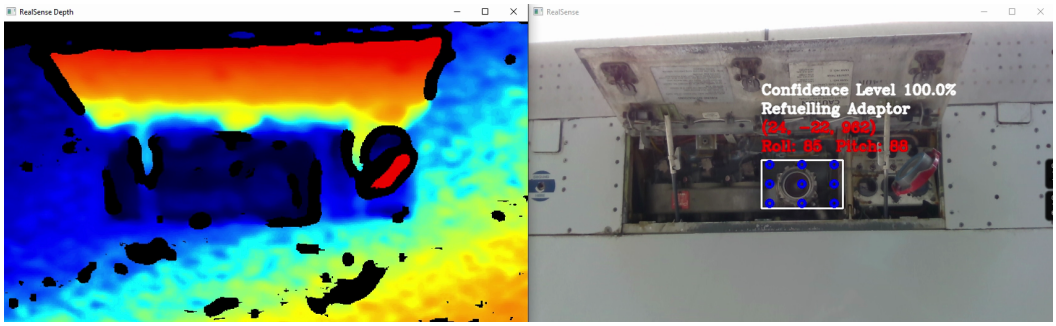


Figure 5. Real-Time Detection in RGB and Depth Stream on Boeing 737-400.

5. Discussion

In this study, we found that a custom neural network, utilising techniques such as Compound Scaling, Neural Architecture Search, and Inverted Residual Block, performed better than pre-trained models, such as EfficientNet-B0, VGG-16, and ResNet-18. The advantage of using a custom neural network is that its architecture is tailored to the specific problem at hand, allowing it to learn more relevant features and improve its performance. Additionally, a custom neural network was designed to be more efficient in terms of computational resources, and memory usage and also incorporate domain-specific knowledge for the problem.

In terms of optimisation, we found that Stochastic Gradient Descent (SGD) performed better than Adam and Rmsprop. This is likely due to the fact that SGD has the ability to escape saddle points or local minima more efficiently, requires less memory to store historical gradients, and can perform well even when the data are noisy or sparse. Additionally, SGD is more robust to the choice of hyperparameters and does not require tuning of the learning rate as frequently as Adam or Rmsprop.

We also found that using a smaller learning rate of 10^{-5} was better than using larger learning rates such as 10^{-4} or 10^{-3}. This is because smaller learning rates allow the optimiser to make smaller updates to the model's parameters, which helps the model converge to the optimal solution more gradually and smoothly, and avoid overshooting the optimal solution. However, it is worth noting that the best learning rate varies depending on the problem and the specific architecture of the model, and it is always best practice to try different learning rates and observe how the accuracy of the model changes with each one. Additionally, using a learning rate schedule or decay helps the model converge more quickly and efficiently, and also helps avoid over-fitting.

6. Conclusions

It has been presented that domain randomisation is a practical technique to reduce the reality gap between the real world and the synthetic environment. The neural network has been trained to accomplish the refuelling adaptor detection task using a hybrid dataset. By carefully manipulating the parameters to generate the synthetic dataset, the domain randomisation method pushes the neural network to discover the fundamental features of the object. By fine-tuning the custom-designed neural network and training it on the hybrid dataset, it has been demonstrated that the resulting model outperformed pre-trained neural networks and led to an improvement in the performance achieved by using the real dataset alone. Using a hybrid dataset to reduce the reality gap is an advantageous strategy to leverage the strength of domain randomisation.

7. Future Work

Future work will focus on reducing the training time by reducing the size of the neural network and manipulating the parameters of the layers. By fine-tuning the custom-designed neural network training time, the computational power and total reaction time of the detection algorithm will be reduced gradually.

Author Contributions: Conceptualisation, S.Y. and Z.A.R.; methodology, S.Y.; software, S.Y.; formal analysis, S.Y.; draft preparation, S.Y.; review and editing, S.Y. and Z.A.R.; supervision, Z.A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This paper does not report research that requires ethical approval.

Informed Consent Statement: Consent to publish statement is not required.

Data Availability Statement: The data that support the findings of this study are openly available in Cranfield Online Research Data at https://cord.cranfield.ac.uk/articles/dataset/AircraftRefuellingAdaptorLocalisation_v3i_coco/20445579/2 (accessed on 15 August 2022).

Conflicts of Interest: None of the authors reports any conflict of interest for this research.

Abbreviations

SGD	Stochastic Gradient Descent
CNN	Convolutional Neural Network
NAS	Neural Architecture Search
VGG	Visual Geometry Group
HDRI	High Dynamic Range Imaging

References

1. Blender—A 3D Modelling and Rendering Package. Available online: https://www.blender.org (accessed on 20 May 2022).
2. Unreal Engine—The Most Powerful Real-Time 3D Creation Tool. Available online: https://www.unrealengine.com (accessed on 28 May 2022).
3. Unity—Real-Time Development Platform. Available online: https://unity.com (accessed on 16 May 2022).
4. Butler, D.J.; Wulff, J.; Stanley, G.B.; Black, M.J. A naturalistic open source movie for optical flow evaluation. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 611–625.
5. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2758–2766.
6. Gaidon, A.; Wang, Q.; Cabon, Y.; Vig, E. Virtual worlds as proxy for multi-object tracking analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4340–4349.
7. Handa, A.; Patraucean, V.; Badrinarayanan, V.; Stent, S.; Cipolla, R. Understanding real world indoor scenes with synthetic data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4077–4085.
8. Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4040–4048.
9. McCormac, J.; Handa, A.; Leutenegger, S.; Davison, A.J. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. *arXiv* **2016**, arXiv:1612.05079.
10. Müller, M.; Casser, V.; Lahoud, J.; Smith, N.; Ghanem, B. Sim4cv: A photo-realistic simulator for computer vision applications. *Int. J. Comput. Vis.* **2018**, *126*, 902–919. [CrossRef]
11. Qiu, W.; Yuille, A. Unrealcv: Connecting computer vision to unreal engine. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 909–916.
12. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for data: Ground truth from computer games. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 102–118.
13. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In Proceedings of the IEE Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243.
14. Tsirikoglou, A.; Kronander, J.; Wrenninge, M.; Unger, J. Procedural modeling and physically based rendering for synthetic data generation in automotive applications. *arXiv* **2017**, arXiv:1710.06270.
15. Zhang, Y.; Qiu, W.; Chen, Q.; Hu, X.; Yuille, A. Unrealstereo: Controlling hazardous factors to analyze stereo vision. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 228–237.
16. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference On Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.
17. Sadeghi, F.; Levine, S. Cad2rl: Real single-image flight without a single real image. *arXiv* **2016**, arXiv:1611.04201.

18. Hinterstoisser, S.; Lepetit, V.; Wohlhart, P.; Konolige, K. On pre-trained image features and synthetic images for deep learning. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
19. James, S.; Davison, A.J.; Johns, E. Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In Proceedings of the Conference on Robot Learning—PMLR, Mountain View, CA, USA, 13–15 November 2017; pp. 334–343.
20. Zhang, F.; Leitner, J.; Ge, Z.; Milford, M.; Corke, P. Adversarial discriminative sim-to-real transfer of visuo-motor policies. *Int. J. Robot. Res.* **2019**, *38*, 1229–1245. [CrossRef]
21. James, S.; Johns, E. 3d simulation for robot arm control with deep q-learning. *arXiv* **2016**, arXiv:1609.03759.
22. Peng, X.; Sun, B.; Ali, K.; Saenko, K. Learning deep object detectors from 3d models. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1278–1286.
23. Dwibedi, D.; Misra, I.; Hebert, M. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1301–1310.
24. CLA-VAL 340AF, Pressure Fuel Servicing Adapter. Available online: https://cla-val-europe.com/en/product/cla-val-340af-pressure-fuel-servicing-adapter/ (accessed on 11 February 2021).
25. Department of Defense, Defense Standardization Program. Available online: https://www.dsp.dla.mil/Policy-Guidance/ (accessed on 4 April 2021).
26. Department of Defense, Defense Standardization Program Procedures. Available online: https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodm/412024m.pdf (accessed on 4 April 2021).
27. Google, The Size and Quality of a Data Set. Available online: https://developers.google.com/machine-learning/data-prep/construct/collect/data-size-quality (accessed on 18 October 2022).
28. Altexsoft, Preparing Your Dataset for Machine Learning. Available online: https://www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better/ (accessed on 12 December 2021).
29. SkyGeek, Military Standard Adapter. Available online: https://skygeek.com/military-standard-ms24484-5-pressure-adapter.html (accessed on 10 May 2022).
30. Yildirim, S. Autonomous Ground Refuelling Approach for Civil Aircrafts using Computer Vision and Robotics. In Proceedings of the IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 3–7 October 2021. [CrossRef]
31. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the CoRR, Zurich, Switzerland, 6–12 September 2014.
32. NVIDIA, What Is Synthetic Data? Available online: https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data (accessed on 19 October 2021).
33. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946.
34. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet: Platform-Aware Neural Architecture Search for Mobile **2018**. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
35. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015. [CrossRef]
36. Meyes, R.; Lu, M.; de Puiseau, C.W.; Meisen, T. Ablation Studies in Artificial Neural Networks. *arXiv* **2019**, arXiv:1901.08644. Available online: http://xxx.lanl.gov/abs/1901.08644, (accessed on 27 February 2021).
37. Ponte, H.; Ponte, N.; Crowder, S. Synthetic data for Blender. Available online: https://github.com/ZumoLabs/zpy (accessed on 2 August 2022).