Cut-and-Paste: Synthetic Image Synthesis by Inserting Object into Image using Deep Convolutional Neural Network

Student Name: Adam Read
Supervisor Name: Neelanjan Bhowmik
Submitted as part of the degree of MSci in Computer Science and Mathematics to the Board of Examiners in the Department of Computer Sciences, Durham University

Abstract—The cluttered and complex nature of x-ray baggage imagery combined with the time-consuming nature of manual threat screening highlights the need for computer-vision-based systems to support human operators. Synthetic x-ray imagery can provide screeners with additional examples and facilitate the production of accurate object detection models. This project aims to explore the use of deep-learning-based composition networks to create this data. We investigate the use of a spatial transformer to predict the threat object location, scale and pose and apply a conditional GAN to predict the insertion appearance. The most accurate composition model produced can generate realistic composite x-ray imagery with consistent appearance and geometry - a Cascade R-CNN detector trained on a hybrid dataset containing our composite images achieved an mAP of 75.4%. Although our results do not improve upon a baseline model trained solely on real data, we highlight the potential of such methods and motivate continued research into deep-learning-based x-ray image composition with a robust and rigorous investigation into the research question.

Index Terms—Computer vision, image generation, neural nets, object recognition

1 Introduction

As we become increasingly globally interdependent, it is clear that international travel plays a growing role in keeping the world connected. Industry estimates prior to COVID-19 suggest that global air transport demand will triple between 2020 and 2050 [62], highlighting the need for robust systems that ensure the safety of passengers and staff. X-ray scanners provide a non-intrusive method of screening luggage, cargo containers, and people for prohibited items such as guns and knives; it is a technique widely adopted by the travel industry and the broader security community. Using x-ray imagery, we can accurately identify the presence of threat items and provide alerts to take the appropriate action.

As objects pass through x-ray scanners, they absorb varying amounts of radiation (depending on the density and thickness of the object), and two plate-like detectors are used to measure the position and intensity of radiation after it has passed through the object. By inserting a filter to block low-energy x-rays between the detectors, we can compare their outputs to construct a projection as a two-dimensional false-colour image depicting object density, position and thickness. Traditionally, a trained human operator will examine this image for the subsequent detection of prohibited items. However, due to the cluttered nature of baggage packing and, consequently, x-ray imagery, this is a challenging and laborious task; studies suggest that humans only achieve an 80-90% accuracy [45]. Caused in part by slow security scanning, it is not uncommon for lengthy airport queues to create delays or cancellations, causing a substantial emotional and financial burden on customers, as seen recently at Manchester Airport [6]. Furthermore, performance can be affected by a range of external factors, from exhaustion to simple distractions. These issues motivate the development of systems to support human operators and further ensure the safety of passengers.

Figure 1: X-ray composition example.

One such technique employed in airports worldwide is threat image projection (TIP). When in use, threat items are superimposed into randomly selected x-ray baggage images, artificially increasing the frequency with which screeners are required to detect threat items. The position within the bag is randomly sampled (then resampled if the projected image lies outside the bag), and some blending coefficient will be applied to improve the appearance of the images [47, 60]. This allows operators to receive frequent detection feedback, which is otherwise missing due to the rarity of natural occurrences. Moreover, TIP can be used as a performance measure. It is required by EU regulation that screeners who do not achieve a minimum accuracy must undergo retraining and only resume screening after passing an x-ray image interpretation test [14]. Whilst the benefits of TIP are evident, they are heavily reliant on the quality of the synthetic threat images. If synthetic images are not representative of real threat images, they cannot be used as a reliable performance metric, and their usefulness as a training tool is dramatically reduced. Despite this, Porta et al. [52] found that 34% of TIP images were unrealistic and could be easily identified as synthetic

by trained screeners. The study suggested the consideration of image characteristics to improve the efficacy of TIP. However, creating a general statistical model to consider such characteristics is challenging not only due to the considerable variation in the baggage and threat items but also the false-colour projections of different x-ray manufacturers. This challenge is not unique to x-ray image projection. Deep learning frameworks have seen impressive success in similar domains [4, 37, 67], evidencing their potential to produce realistic synthetic x-ray imagery.

In addition to systems such as TIP, which aim to improve the reliability of a screener, we can create systems that aim to lessen the difficulty of threat detection. Object detection systems, which predict the presence and bounding box of a threat item, are a natural choice. However, a fundamental limitation restricting computer-vision-based x-ray threat detection performance is a lack of widely-available threat images; the most comprehensive dataset, SIXRay [44], contains just 8,929 positive images split over six threat classes. Moreover, due to the rarity of natural threat object occurrence, producing such imagery requires scanning and annotating many images manually, which is a very time-intensive task. As such, datasets have insufficient diversity in object coverage and construction. When faced with limited training data, it is common to apply data augmentation methods such as image translation, rotation, flipping, or rescaling to increase the size and diversity of the dataset artificially. Webb et al. [70] show that although data augmentation can have limited success, improving from a baseline mAP of 85.2% to 85.8% using a Free Anchor [77] architecture on SIXray [44], it is not an adequate solution for x-ray screening tasks. The augmented images still lack diversity in the variation of the pose, scale and item construction of threat objects. In similar tasks, image synthesis has been shown to improve the performance of detection models reliably [17–19], highlighting the need for experimentation with such techniques in the x-ray domain.

This work focuses on the generation of synthetic yet realistic, prohibited x-ray security imagery using deep learning. To our knowledge, this is the first work to explore the use of deep learning techniques for such composite x-ray imagery. Moreover, by training using a diverse set of both synthetic and real imagery, we produce novel results to explore the efficacy of deep learning based composition for the production of accurate threat object detection networks.

1.1 Background

Abstractly, image composition networks aim to insert some foreground object into a background such that the resultant image is realistic. It follows that to produce realistic synthetic imagery, we must consider consistency between images in two key areas: colour and geometry. For an inserted image to be colour consistent with a background, it should look natural in its chosen position, matching the background lighting and tones. Whereas, for geometric consistency, we must place the foreground object in a realistic location, a point in the scene where the object could naturally occur. Furthermore, the foreground object must be scaled, rotated and otherwise transformed to the visual context of the chosen location. The better we perform in these two domains, the better the composite image.

We will use a Spatial Transformer Network (STN) [30] to provide geometric consistency to our image. A Spatial transformer uses a dynamic mechanism to learn an appropriate transformations for each input sample; this can include scaling, cropping, rotations, as well as non-rigid deformations. Although STNs were initially proposed to provide spatial invariance for convolutional operations, they have been shown to learn realistic transformations for the placement of objects within a scene [4, 37, 67]. After transformation, we use a conditional Generative Adversarial Network (cGAN) [46] to insert the threat image into the bag. Conditional GANs have a similar network structure to traditional GANs [25]. Trained jointly, GANs consist of two components; a generator that produces synthetic imagery and a discriminator trained to identify images as either natural or synthetic. In training, the generator learns to create compositions able to fool the discriminator, if the generator can do this it has successfully mimicked the input data distribution. Although GANs can generate compelling imagery, often, we have very little control over the image generated. Therefore, to better inform the generation, we can provide the model with additional auxiliary information; this information is said to condition the GAN, extending it to a conditional GAN.

When composing two images, we attempt to learn a mapping to produce a composition indistinguishable from real data. However, this does not require that the background and foreground are linked in any meaningful way and can often lead to mode collapse, in which every output becomes identical. First introduced by Zhu et al. [80], ensuring our networks are cycle consistent requires that we measure the quality of our network by both the produced synthetic image and the ability of the network to decompose back into a foreground and background. By linking the composition and decomposition, we can often produce a higher quality composite. In this paper, we leverage that idea by applying a cycle consistency loss to the transformation and insertion of threat objects.

To training a composition network, we consider two types of threat datasets: paired and unpaired. In a paired dataset, as well as the composed imagery, we have individual scans of the threat item and the bag. Using the composed image, we can define the success of the network at every stage accurately; it is simple to measure the geometric and colour consistency using the ground truth transformation and insertion. Whereas in an unpaired dataset, we only have examples of the composition. As the correct composition can vary significantly between each luggage-threat pair, using a composed image to evaluate the composition of a different pair is not reliable. Therefore, we use an image completion network to synthesise paired imagery from our unpaired dataset. We remove the threat item from the composed image using manual annotations, then apply a self-supervised inpainting network [50] to synthesise the cut-out bag area. The self-supervised inpainting network is also a cGAN and can be trained similarly to the insertion network.

In this paper, we will use a composition network to generate synthetic x-ray threat imagery. We will examine the use of both paired and unpaired images and evaluate the efficacy of both approaches. We use the Durham University x-ray image dataset dbf3 [7] to test the application for unpaired images and we create our own paired dataset consisting of 998 examples. Furthermore, we produce novel results compare the use of Dice Loss, Tversky Loss and Focal Tversky Loss in the training of a cycle consistent spatial transformer. Finally, we will use Feature Selective Anchor-Free (FSAF) [79] and Cascade R-CNN (R-CNN) [10] models to evaluate the success of our approach and explore the use of such images to improve the accuracy of object detection networks.

1.2 Project Objectives and Achievements

The research question asked in this report is: *Can contemporary advances in deep-learning-based composition networks be successfully applied to the domain of x-ray imagery?* When considering how to answer this question, we determined the following objectives:

The minimum objectives of this project were to adapt and apply the image composition network proposed by Azadi et al. [4] for the composition of x-ray baggage and threat items using the unpaired dbf3 dataset. In addition, we aimed to evaluate the qualitative success of this network with a visual comparison between the synthetic imagery generated and traditional TIP methods. These objectives were proposed to develop a strong understanding of prior work and generate a foundational model from which we can adapt.

Our intermediate goals were to extend our work using paired data and formalise our evaluation method using quantitative results. We aimed to create our own paired x-ray image dataset with which we could produce a detailed comparison between paired and unpaired data for x-ray composition. To complete a quantitative evaluation of our work, we aimed to explore the success of multiple object detection networks from different families (that is, different architectural approaches) trained both on synthetic and real imagery. We sought to understand the full potential of our baseline approach and provide novel object detection experimentation. Moreover, we aimed to improve upon the state-of-the-art of x-ray object detection models trained using synthetic composed images, a baseline mAP of 81% [7].

The advanced objectives were to adapt the original model by implementing and evaluating different loss functions better suited to the x-ray domain. Furthermore, the current state-of-the-art detector on dbf3 achieves an mAP of 88% [22]; we aimed to improve the performance of object detection models on threat imagery against using some combination of real and synthetic imagery. With these objectives, we aimed to produce techniques to advance the state-of-the-art performance of models within the composition and the object detection domains.

Other than improving the performance of object detection networks, we successfully achieved all basic, intermediate, and advanced objectives. To produce a qualitative evaluation of our model, we used Cascade R-CNN and FSAF object detection networks. Moreover, we investigated the use of Dice, Tversky and Focal Tversky loss to adapt the architecture proposed by Azadi et al. [4]. Our best model was used to produce a hybrid dataset able to facilitate an mAP of 75.4% using Cascade R-CNN and 68.6% using FSAF against dbf3. This did not improve on the baseline model trained solely on real data or the models trained using TIP imagery.

2 Related Work

In the field of computer vision, image composition refers to the process of combining a foreground object with a background image to generate a realistic-looking composite. Using image composition, we can generate novel and diverse amalgamations of foregrounds and backgrounds to increase available training data for deep learning models substantially. For this, when training deep learning models, it is clear that contextual information within images can dramatically influence performance [66]. As discussed in 1 this requires attention to geometric and colour consitency.

2.1 Colour Consistency

Early work in the field takes a naive, colour-consistency-based approach; composites consist of foreground images inserted into a background at predetermined positions using simple blending techniques. Seminal work in the field uses linear interpolation with alpha matting to combine foregrounds and backgrounds [53]. Similarly, later studies merge Laplacian pyramids of images then upsampled to reconstruct a full-scale composite [8] and apply Laplacians to solve the Poisson equation numerically for each colour channel independently [51]. Despite their importance, blending approaches such as these are fundamentally flawed in image composition. If we consider inserting a red object into a yellow background using blending, the red object must be, in some way, tinted yellow. Regardless, it is realistic to have a red object on a yellow background without any tint - it may even look unrealistic otherwise. Simply mixing the images can unnecessarily remove important details and affect the quality of synthetic imagery.

First introduced by [55] for colour correction, research began reviewing the efficacy of colour shifting as a simple way to improve the colour consistency between a foreground object and a background image. In colour shifting, we generate histograms of the colours present in two images then shift the histogram of one image to match the other. Lalonde and Efros [35] investigate this techinque by shifting the foreground to match either the background image palette, a global palette summarising all images in the dataset, or a combination of both. They evaluate the performance of different techniques by assigning a realism score using the χ^2-distance metric for every image in the test set, constructing a Receiving Operating Characteristic curve (ROC) and calculating the Area Under the Curve (AUC). Using [55] as a baseline, simple colour shifting techniques achieve an ROC AUC of 0.66. In comparison, [35] achieves an ROC AUC of 0.81 using a combination of local and global palettes, showing the importance of learning compositions using the entire domain of background images, not just individual images. Building upon this work, [72] consider not only colour statistics but a multitude of key statistics that can affect the realism of photo composites, such as luminance, colour, temperature, saturation and local contrast. Xue et al. [72] apply the colour shifting principle in each statistic to produce a composite. Furthermore, [63] use colour shifting as an extension to the work in [8]: applying colour shifting at each level in a pyramid, then upsampling to produce a composite.

As with many problems within computer vision, deep neural network architectures dominate contemporary image composition approaches. Comparable to the use of global colour palettes [35], deep learning can improve upon traditional techniques by learning complex relationships across entire datasets rather than using handcrafted heuristics that often consider images to be independent. Seminal in its approach, [20] applies a CNN variant shown to have high performance in image classification tasks, Visual Geometry Group (VGG) [61], to learn a set of representations from each stage of filtering in the network. Gatys et al. [20] reconstruct the image with altered styling by applying gradient descent on a white noise image and jointly minimising for the original image and the style representation. Later, [81] use a VGG to distinguish natural images from generated composites. By iteratively optimising the parameters for the colour-shifting approach using the loss produced by composite realism through the VGG, [81] improves upon the colour consistency of composites. Several methods built upon this work, applying VGGs similarly. In [21],

researchers explore optimisation of spatial style, colour and scale parameters, whereas [68] optimises by training a pair of decoder CNNs independently from the VGG to reconstruct the composite.

Generative Adversarial Networks (GANs) [25] are deep learning models that have shown impressive results on image generation, and by extension, composition, tasks. In [64], researchers first explore their use with a source and style image input to a basic GAN with a Variational Auto-Encoder (VAE) [33] generator and CNN discriminator to generate colour consistent foreground objects. Whilst this produces impressive initial results, the first significant breakthrough in GAN-based style transfer came in [29], which proposes the pix2pix architecture. Isola et al. [29] use conditional GANs with a U-Net [58] generator and a PatchGAN-based fully convolutional discriminator [15]. This architecture addresses two key issues; the U-Net generator contains skip connections between layers that share much more low-level information through the network, and the PatchGAN discriminator classifies patches of images reducing the blurriness produced from traditional L1 or L2 losses. However, this approach is not unpaired; thus, a substantial amount of labelled training data is required to train the pix2pix architecture [29]. Furthermore, there is a lack of diversity and it is hard to generate high-resolution outputs.

In previous attempts to produce an unpaired translation model, often the model would fail with mode collapse; all forward translations would collapse into the same image. Zhu et al. [80] propose a cycle-consistent unpaired translation model to solve this, cycleGAN. For a model to be cycle-consistent, it must be able to learn to translations both forward into a new domain and backwards into the original domain. Moreover, the translators between the two domains, $G : X \rightarrow Y$ and $F : Y \rightarrow X$, should be bijective inverses; that is, $F(G(X)) = X$. Applying cycle consistency allows cycleGAN [80] to achieve impressive results, vastly outperforming all previous unpaired image translation models. Despite this, as [80] enforces cycle consistency using a L1 loss between the original image and the backwards translation, only low-level information can propogate through the network. This can cause blurriness in the resultant image and reduces the efficacy of the network when translating between domains with imbalanced information levels such as sketch to image. Later works propose alternative loss functions to improve the quality of images whilst maintaining cycle consistency. Researchers explore the use of classical image quality loss functions such as FSIM [13], high-level feature comparison using a U-Net architecture [13] and adversarial loss functions realised by a discriminator to classify the original and translated images [78]. However, instead of simply loosening the cycle consistency constraint, the most successful recent approaches maximise the information provided to the translation network. Park et al. [49] use a patchwise contrastive loss to maximise the relationship between the input and output. Furthermore, the current state-of-the-art unpaired translation model [65], uses self-attention to separate the foreground and background. This separation allows their AttentionGAN model [65] to focus on the translation of the foreground, further improving the consistency between translations.

Contemporary paired translation models address the second shortcoming of such approaches, [28] generates diverse synthetic images by combining a content code with multiple style codes to produce multimodal outputs. Moreover, pix2pixHD [69] produces high-quality synthetic imagery by considering a feature matching loss with multiscale generators and discriminators. Furthermore, [57] produces multimodal, unpaired high-resolution images. The pixel2style2pixel (pSp) architecture [57] generates imagery using a novel pyramid-based encoder architecture. Style vectors are extracted at differing pyramid scales and inserted directly into a pre-trained StyleGAN architecture [32], allowing latent space manipulations that produce high-quality synthetic imagery that outperforms other contemporary methods. In a study performed by [57] with over 8,400 human comparisons between pSp and other state-of-the-art paired methods, over 90% of users favoured the image produced by pSp, evidencing its excellent performance.

2.2 Geometric Consistency

The potential of composition networks to improve the performance of object detection networks was first uncovered in [18], which introduces a cut-and-paste-based method. Their network randomly inserts foreground objects into backgrounds to produce composite imagery. Random insertion improved object detection performance by 21% on benchmark datasets; however, it does not allow object detection networks to learn high-level relationships. To produce synthetic imagery capable of further improving the performance of object detection algorithms, we must consider the position at which we insert an object. In [24], a RANSAC algorithm estimates the semantic map of support surfaces to find regions suitable for object placement. Georgakis et al. [24] insert objects onto these support surfaces to generate synthetic imagery. Through this, [24] improves the performance of a Faster R-CNN [56] detector from 82.5% to 85.0% on the GMU-Kitchens dataset [23]. However, this approach is not general enough; foreground objects do not always occur on support surfaces.

More advanced cut-and-paste networks learn object-specific predictions rather than general insertion locations for any object. Dvornik et al. [17] predict the bounding box of objects by training an object detection network on scenes after removing objects with image inpainting. When composing, they insert the foreground object into the predicted bounding box of the scene. Later, [19] proposes InstaBoost; their architecture learns an appearance consistency heatmap of object locations. Using this map, InstaBoost samples feasible positions during composition and conducts crop-paste data augmentation. The most recent cut-and-paste method introduces PlaceNet [76]. Zhang et al. use a encoder-decoder generator to combine representations of the foreground and background then decode a combination of these representations to bounding box predictions [76]. As with other adversarial networks, this is trained using a discriminator to classify the likelihood that predicted bounding boxes are valid. An interesting contribution of [76] is to synthetically create paired data by extracting the foreground from unpaired images and applying an inpainting network to synthesise a background. This allows traditionally paired networks to extend to the unpaired domain. A qualitative evaluation of images generated using PlaceNet found that 76.4% of respondants found the placement to be plausible. Despite their success in object placement, these techniques can apply minimal augmentation to the foreground object. None of the cut-and-paste algorithms can rotate, flip, or otherwise transform a foreground beyond placement and scaling. This limitation significantly reduces the applicability of such techniques in object detection domains with high variation in pose and construction.

To increase variety in composition construction, researchers explore the use of spatial transformers [30] in adversarial networks. Using a spatial transformer as the generator, ST-GAN [37]

can predict realistic warps beyond the location and scale of the foreground, such as cropping, rotations, and other non-rigid deformations. A discriminator assesses these transformations producing an adversarial loss determining the realism of transformations. An extension to ST-GAN, [67] finetunes composition to a task, such as object detection or image classification. Using a pretrained target network in the architecture, both the adversarial and target network losses are combined to determine the success of a composition. Training on images in [67], the mAP of a Faster R-CNN model improves from 86.3% to 89.8%.

2.3 Geometric and Colour Consistent Composition

All the networks discussed thus far can either style or spatially transform the foreground; however, it is clear that a network capable of producing realistic imagery, regardless of the image domain, must learn both. The seminal work in this task, [11], proposes Geometric and Colour Consistent GAN (GCC-GAN). The generator in GCC-GAN [11] takes a triplet consisting of the foreground object, foreground mask, and background as input to two branches used to solve the two tasks. Geometric consistency is considered in the first branch; an STN is used to warp the foreground mask as in [37]. The second branch uses the network proposed in [81] to adjust the style of the image. The composite image is produced by combining the mask and foreground style using the Hadamard product then superimposing onto the background. Later work removed the need for foreground masks, [75] introduces Spatial Fusion GAN (SF-GAN). Rather than warping a mask of the foreground, SF-GAN [75] uses an STN to predict the geometric warp of the foreground image directly. The translated foreground is then superimposed onto the background and fed to a GAN with an adversarially supervised weak cycle consistency to improve the colour consistency. Both methods require paired data to learn compositions; thus, an extensive data collection task is required for practical applications.

Researchers later explore an extension to the work in [75], HIerarchical Composition GAN (HIC-GAN) [74]. The proposed network is seminal in its ability to learn multiple-object image composition. For this, HIC-GAN [74] uses three STNs to predict the translations of objects using a vector representation learned by a DenseNet architecture [27]. Then, object occlusions are calculated using a weighted learned sum of all possible occlusions produced using a softmax over a fully connected layer. Finally, the image is superimposed as in [75] and undergoes the same translation process with further supervision using an attention-based loss to enforce cycle consistency. Whilst this multiple-object composition work is an area for future research; we only consider single-object x-ray composition in this paper. The quality of single-object composition between SF-GAN [75] and HIC-GAN [74] is slight; in a survery run by [74] 38% of respondents classified compositions produced by HIC-GAN [74] as realistic compared to 35% of respondents for SF-GAN [75].

Finally, [4] proposes the state-of-the-art model in the field, compositionGAN. Here, as before, researchers leveraged the idea that a successful composition of two images should be realistic in appearance and should be decomposable into individual objects. Similarly to previous works, an STN is used to warp the foreground image. Then, the background and warped foreground are fed to a conditional GAN to produce colour consistency. Aziz et al. [4] improve upon previous work by enforcing cycle consistency for the geometric warp, not just the appearance. This cycle consistency is supervised using an adversarial and L1 loss. This architecture allows compositionGAN [4] to produce impressive composites; 84% prefer composited images generated by compositionGAN [4] to counterparts generated by ST-GAN [37]. Moreover, compositionGAN can extend to the inpaired domain using a similar technique to that proposed in [76]. They use the inpainting network proposed in [50] to significantly increase the amount of training data available to the network, a vital strength in data starved domains such as x-ray imagery.

2.4 X-Ray Image Composition

There has been little work attempting to produce synthetic composite x-ray imagery. Initial attempts focused on traditional threat image projection, in [31] a database was created using a simple two-step process: superimposition of threat items onto a bag followed by image transformation methods such as rotation, flip, zoom, skew and distortion. Building from this, [7] improved the superimposition method by first applying some random rotation to the threat object and then using trial-and-error to randomly insert the threat object until it is entirely inside the bag. After finding the insertion region, the consistency of transparency and contrast between the threat item and background is considered by using a threat threshold to remove high-value pixels from the threat signals.

The most recent paper on x-ray image synthesis, [36], is the first to consider synthesis using an adversarial method. A self-Attention GAN (SAGAN) [40] generates threat objects with various postures. Then, a similar GAN is applied to construct the bag with the threat object, effectively producing realistic and natural imagery. Through this process, researchers improved upon the state-of-the-art object detection results for x-ray imagery using a CNN to identify threat objects with 98.4% accuracy. In this paper, we will be extending research in this field by considering the application of GANs on x-ray image synthesis through composition.

In this paper, we will be extending research into synthetic x-ray image composition by exploring the use of compositionGAN [4] in the x-ray domain. To our knowledge, we are the first paper to consider adversarial composite x-ray image synthesis, a method that has produced impressive results in other, similar fields allowing us to produce novel results and conclusions to evaluate the efficacy of such networks and motivate the need for further investigation. Moreover, we run novel experiments exploring the use of Dice loss, Tversky loss and Focal Tversky loss to supervise the cycle consistency for STNs, allowing us to evaluate their efficacy in not only the x-ray domain, but image composition as a whole.

2.5 X-Ray Object Detection

Accurate threat item detection is an important task undertaken primarily with human supervision. By creating systems able to identify threat items, we can reduce the pressure on human operators to improve the safety and security provided by x-ray scanning. Early work focused on traditional computer vision methods, primarily Bag of Visual Words (BoVW) models for threat detection [5, 34, 41]. Seminal research [5, 41] suggests such techniques in isolation achieved poor results; however, by finetuning models with information gathered in multiple views or colour spaces, researchers significantly improve the performance of models. For example, in [41], the inclusion of multiple views

improved the precision of a BoVW detector from 71.4% to 95.7%. Later, [34] explores the use of numerous feature point detectors, descriptors, and classifiers, their best model achieving a 94% accuracy using FAST-SURF feature points with an SVM classifier.

Recent approaches have tested contemporary CNN architectures for x-ray object detection. Despite their impressive performance in other domains, deep CNNs traditionally require a large amount of training data to facilitate the construction of a complex end-to-end feature extraction process. Within the context of x-ray security screening, the limited availability of images causes a significant bottleneck for the performance of such models. Thus, [2] utilised transfer learning. Akçay et al. [2] finetuned models pretrained on ImageNet [16] for x-ray baggage screening, achieving an mAP of 98.4% on GDXRay [42]. To inform later research, [43] compares traditional computer vision techniques against pretrained modern CNNs. As [43] shows, deep learning techniques can outperform traditional computer vision reliably, improving the mean detection accuracy by 0.9%. Continuing to explore CNN-based methods, [3] perform exhaustive comparisons of various architectures and transfer learning techniques. [3] shows a Faster R-CNN model achieves a 88% mAP on dbf3. Moreover, [22] tested the robustness of deep learning architectures on an adversarial dataset. Through their investigation, they found that a RetinaNet model [38] produced a false positive rate of just 5%, highlighting its reliability.

To create more comprehensive x-ray datasets, [44, 71] introduced OPIXRay and SIXRay, the latter containing over 1,000,000 images. In their creation, researchers produced baseline results. For OPIXRay, [71] proposes a De-Occlusion Attention Model (DOAM), which achieves a baseline mAP of 82.41%. In comparison, [44] produces a maximal mAP of 79.56% on SIXRay using a DenseNet architecture [27]. The most recent advances continue to investigate techniques to increase the training data available synthetically. Bhowmik et al. [7] train a detector using TIP imagery; however, they cannot improve upon the detection performance. The best performing Faster R-CNN [56] model, trained using both real and synthetic data, achieves an mAP of 81%. Finally, [70] uses standard data augmentation techniques such as flipping, cropping and rotation. The augmentations are evaluated using a Free Anchor [77] and a Cascade R-CNN [10] models, improving the state-of-the-art detection on SIXRay and OPIXRay to 90.9% and 85.8%, respectively.

Beyond image composition, we will explore the use of FSAF and Cascade R-CNN models trained using a combination of real imagery, from the dbf3 dataset, and synthetic composition imagery. Similar to previous work, [7, 70], this allows us to artificially increase the size and diversity of training data. We will compare our results to that of [7] to evaluate the use of composite imagery in x-ray detection and the current state-of-the-art result produced in [3].

3 Methodology

In this section we will detail the workflow necessary to produce an effective x-ray image composite. After considering the implementation tools required, we will discuss the use of spatial transformers (3.2), conditional generative adversarial models (3.3) and image inpainting networks (3.5) to provide an overview of the network architecture proposed by [4] (3.6). We also motivate our novel experimentation with various loss functions (3.8) to supervise the spatial transformer and discuss the object detection models chosen to evaluate our composites (3.7). Finally, we will consider the datasets used for the training and evaluation of our composition and object detection models, including the creation of a paired dataset for this work.

3.1 Implementation Tools

We chose to use PyTorch [54] to produce our composition network. As the industry standard, PyTorch is well documented and supported. Furthermore, it is fast, flexible and provides an easy to use framework for deep learning. To produce quantitative results, we used object detection models from the prebuilt toolbox developed by MMDetection [12]. MMDetection supports a variety of advanced object detection networks with prebuild models that train at speeds faster or comparable to other codebases. Moreover, to implement the data augmentation methods discussed in this project, we use the toolbox provided by Albumentations [9]. Finally, to annotate our images for the production of our unpaired dataset, we used the annotation tool provided by [48].

3.2 Geometric Consistency - STNs

When considering how to create composite imagery, we must apply some transformation to alter the structure and position of the foreground object to fit the visual context of the background. It is important to note that, as x-ray imagery is a projection from 3D to 2D, the use of just rotation, scaling, and translation is not enough. To understand this, consider an object orthogonal to an x-ray detector. Suppose we attempt to compose this object into a background by applying rotation, scaling and translation. The resultant threat item in the composite will still appear orthogonal to the detector. However, this construction is not the only realistic possibility - the pose of our input data is limiting its composition. On the other hand, we must not deform the image to lose crucial structural details. To preserve such information, we limit our model to the class of 2D affine transformations, which preserve parallelism between lines, the ratios of lengths between parallel lines, and colinearity. As well as the aforementioned transformations, we can now include dilation and shearing. Thus, we can mimic a change in perspective. Using affine transformations, the network can retain the structure of an image whilst warping to produce realistic compositions.

All affine transformations can be represented using a 3×2 matrix. It follows that learning an affine transformation can be abstracted to the regressive prediction of six parameters such that the transformed position (x^t, y^t) of an input coordinate (x^s, y^s) is given by,

$$\begin{pmatrix} x^t \\ y^t \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} x^s \\ y^s \\ 1 \end{pmatrix}. \tag{1}$$

We use a spatial transformer network to predict these parameters and apply the subsequent transformation to the input image.

3.2.1 *Localisation Networks*

Spatial transformers consist of three main components: a localisation network, a parametrised sampling grid, and a differentiable grid sampler. The first of these, the localisation network, predicts the parameters for affine transformation. Before prediction, however, the network contains a series of convolutional layers. Intuitively, passing the image through these convolutional layers

allows the network to finetune filters that extract the most pertinent information for transformation. We first propogate through a 7×7 convolutional filter followed by two 5×5 convolutional layers. This initial block increases the number of layers in each map from 3 (representing the RGB colour channels) to 128. The feature map is then passed through further 5×5 kernel convolutional layers until we have reduced to a feature dimensionality of 4×4×128. Between each convolutional layer, we use a max pooling layer and apply a ReLU activation function. After learning this compact representation, it is flattened to a 1×2048 vector and used as input to a regressive layer. A fully-connected deep neural network, the regressive layer passes the vector through a series of neural network layers to learn complex non-linear to predict the parameters of the transformation. In summary, the localisation network allows for a sophisticated association between the input image and the predicted transform to be learned; with such a diverse set of possible transformations in x-ray composition this is crucial.

3.2.2 *Sampling Grids*

Before transforming the image, we produce a parametric sampling grid of equal size to the output image. The sampling grid provides a map connecting the input and output images; each coordinate in the sampling grid defines the relative spatial location in the input image for a sampler to be applied. The input pixel sampled denotes the output pixel at that coordinate. For example, the spatial location defined at each pixel in the identity sampling grid is the same as the coordinate of that pixel in the input image. Therefore, when sampled, the image would remain unchanged. Suppose we apply our transformation to the identity sampling grid using equation (1). With the transformed grid, we can efficiently apply our transformation globally to any image by sampling using the defined relative spatial locations.

3.2.3 *Differentiable Grid Sampler*

The final component, a grid sampler, defines how we should sample from this map. Firstly, we note that it must be differentiable. By enforcing differentiability, we allow standard backpropagation to update the parameters of the localisation network; if our sampler is not differentiable, our model will not learn. After ensuring differentiability, we could naïvely attempt to collect the exact pixel described by the grid sampler. However, transformations often do not map directly to a pixel. Instead, they can lie between several pixels in a non-integer location. Although our sampler could collect the nearest pixel, this may cause transformations to look disjoint and unrealistic. Instead, we use bilinear interpolation. When a transformed pixel must sample from a non-integer pixel location, its value is estimated using a weighted average of the four pixels surrounding it. The distance between each pixel is directly proportional to its weight in the estimation; closer pixels contribute more to the estimation. Using this architecture, the spatial transformer applies a learned transformation identically in each colour channel to produce a resultant composite.

3.3 Geometric Consistency - Transformation Loss

We use a spatial transformer as part of a supervised training problem. As such, we provide the model with input and ground truth data. The spatial transformer will attempt to estimate the correct transformation during training by predicting the ground truth. This estimate is compared with the correct transformation, and the loss between them is calculated. Loss is a measure of the

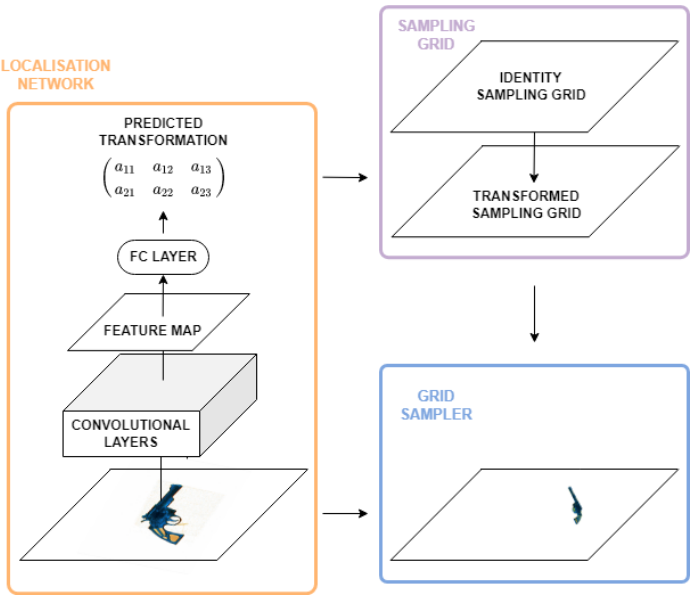

Figure 2: Spatial Transformer Network architecture.

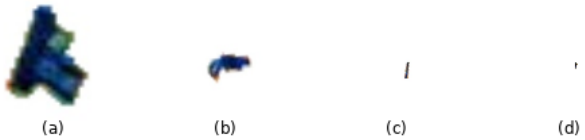

Figure 3: STN predicted transformation after (a) 10 steps, (b) 50 steps, (c) 100 steps and (d) 200 steps.

transformation accuracy; the more similar the output and ground truth are, the smaller the loss between them. As proposed by [4], we can use an L1 loss to supervise this, calculated as the absolute difference between the two images. However, the L1 loss does not produce a balanced representation of the geometric distinctions between a pair of images. For example, consider the L1 loss between two images containing small objects compared with images of large objects. As the large objects use many pixels, a slight inaccuracy in transformation may appear as a substantial L1 loss. Whereas, as the pixels will differ in just a few places, a more significant inaccuracy in transformation could cause a relatively slight change in the L1 loss. As our target transformation reduces the scale of an image, this is a significant problem when learning the correct spatial transformation for x-ray threat items. We found that the model began fit to a local minimum, transforming the threat item to occupy just a few pixels. Whilst this significantly reduces the L1 loss, it does not create realistic transformations. To improve our compositions, we experimented with different loss functions.

When considering a loss function to evaluate the geometric consistency, it is important to note that we only care about the shape and position of the output. Therefore, we can abstract the pixel values, binarising them to represent either the presence or absence of an object. Using this discretised image, we can apply classical image segmentation loss functions invariant to object scale. We experiment with the use of Dice loss, Tversky loss and Focal Tversky loss.

Widely used in set theory, the Dice loss was initially proposed to measure the similarity between two sets. We adapt this idea by defining the two sets as pixels locations of the predicted and ground truth object transformations. We use the proportional

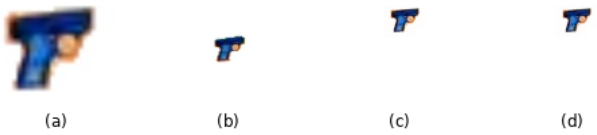

Figure 4: Predicted transformation of (a) threat item by an STN trained using (b) Dice loss, (c) Tversky loss and (d) Focal Tversky loss.

overlap between images as a measure of success. For clarity of notation, we state that True Positive (TP) pixels are those in the overlap, False Positive (FP) pixels are those incorrectly predicted to contain the object, and False Negative (FN) pixels contain the object in the ground truth but not the prediction. The Dice loss is given by,

$$\mathcal{L}_{\text{dice}} = 1 - \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}} \tag{2}$$

The Tversky loss is a generalisation of Dice loss. It allows for the weighting of importance between False Positive and False Negative results by introducing two hyper-parameters, α and β. Generally, we use $\alpha+\beta = 1$ to allow for a loss scaled from 0 to 1. Intuitively, increasing α penalises learning transformations smaller than the ground truth, whereas increasing β penalises learning transformations larger than the ground truth. The Tversky loss is given by,

$$\mathcal{L}_{\text{tversky}} = 1 - \frac{\text{TP}}{\text{TP} + \alpha\text{FN} + \beta\text{FP}} \tag{3}$$

Finally, the Focal Tversky loss is a further generalisation of the Dice loss. We introduce a third hyperparameter, γ, to control the non-linearity of the loss. Intuitively, if $\gamma < 1$ the gradient of the loss is higher, the model is further incentivised to learn when nearing convergence. Whereas, if $\gamma > 1$, the model is forced to focus on harder examples, small transformations which could recieve low Tversky or Dice losses would be penalised greater.

$$\mathcal{L}_{\text{focal tversky}} = \left(1 - \frac{\text{TP}}{\text{TP} + \alpha\text{FN} + \beta\text{FP}}\right)^{\gamma} \tag{4}$$

For our implementation, we use the values $\alpha = 0.3$, $\beta = 0.7$, and $\gamma = 4/3$ based on experimental results shown in [1, 59].

3.4 Colour Consistency - Conditional GANs

We briefly review Conditional GANs before discussing our architecture, as proposed by [4]. Abstractly, GANs are generative models that learn to create synthetic data within a target domain. Such networks use a random noise vector z, a generator G, and a discriminator D. The generator takes noise as input to produce synthetic imagery x, and the discriminator attempts to distinguish between real and fake images. By training these jointly, the generator must learn to produce synthetic images able to fool the discriminator whilst the discriminator learns to classify imagery of increasing realism, further improving the generator. When the generator can reliably fool the discriminator, the model has learned to generate images following an identical distribution to the target. This setup, however, provides minimal control over the image generated, causing issues such as mode collapse where almost every generated image becomes identical. Therefore, we extend to a cGAN architecture by providing the model with information y to condition the generation and control the result. Conditional GANs can be said to learn the mapping between the auxiliary domain and the target domain. In our network, we provide the network with a baggage-threat pair as auxiliary information. The adversarial loss describing this is given as,

$$\begin{aligned}\mathcal{L}_{\text{cGAN}}(G, D) = \mathbb{E}_x[\log D(y, x)] \\ + \mathbb{E}_{y,z}[\log(1 - D(y, G(y, z)))].\end{aligned} \tag{5}$$

We use an L1 loss for further supervision to penalise deviation from the ground truth composite $\hat{x}$ given by,

$$\mathcal{L}_{\text{L1}} = \mathbb{E}_{x,y,z}[||\hat{x} - x||_1]. \tag{6}$$

In our generator, we use a traditional encoder-decoder structure. The encoder compresses the inputs into a low-dimensional latent space producing a concise representation of features; in-essence, we downsample the inputs. To implement our encoder, we propogate through two convolutional layers with a 7×7 and 3×3 kernel respectively, followed by a series of residual neural network (ResNet) blocks. Before motivating the use of ResNet blocks, we first note that when using neural networks, we effectively propagate information about the input down the network as we pass through layers. However, as we deepen our model, the efficacy of this propogation worsens; the quality of information begins to degrade. To improve this process, residual CNNs introduce skip connections. These shortcuts connect shallow and deep neurons directly, helping to pass information through the network with less degradation. Using residual blocks allows for a deeper architecture whilst ensuring that each layer performs at least as well as the previous layer, due to the skip connection joining them. As a result, we can filter the image more effectively. In our network, ResNet blocks are built using two 3×3 convolutional layers with skip connections joining the layers immediately before and after each block. After our compact representation has been learned, we must reconstruct from the latent space to produce a composite image. To produce an output of identical size to the input, we iteratively upsample our representation using nearest neighbour interpolation to double the size at each step. Each time we upsample, a 3×3 convolutional layer is applied to convert from a latent representation into composite imagery and reduce the aliasing effect. Finally, a 7×7 convolutional filter is used on the final pass. After the decoder has generated an image, it is passed into the discriminator.

In our discriminator, we use a PatchGAN classifier as proposed in [29]. To consider the structure of our classifier, we first need to consider the supervision provided by the L1 loss. As has been extensively researched, when using an L1 loss with an encoder-decoder structure, we can capture low-frequency information accurately but struggle to generate high-frequency image attributes [29]. This lack of high-frequency detail can cause the generator to produce blurry images without fine details. Thus, as proposed in [29], we use a classifier to enforce the learning of high-level attributes by using PatchGAN. In order to model high frequencies, it is sufficient to restrict our attention to the structure in local image patches; therefore, PatchGAN classifies on such a scale. Our PatchGAN discriminator runs convolutionally over the entire image using classifying every 70x70 image patch as real or fake. After extracting a patch, 3×3 convolutional layers are iteratively applied, reducing to a classification prediction. By averaging all responses, we produce the overall discriminator output. In [29], the patchGAN discriminator was shown to produce impressive results independent of image size; thus, we do not need to update our discriminator parameters for different input or output

shapes. Classifying smaller patches instead of the entire image also requires fewer parameters allowing for faster training.

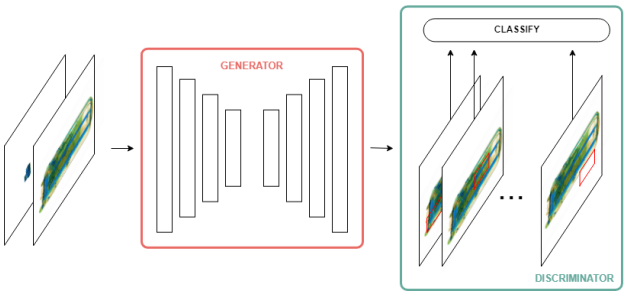

Figure 5: Conditional GAN architecture.

3.5 Extension to Unpaired - Self-Supervised Inpainting

In both our geometric and colour consistency networks, we use ground truth images to supervise the performance of our network. Therefore, we require paired imagery connecting individual threat objects, bags, and their composition to train our network. In the x-ray domain, there is a significant lack of such data; many publically available datasets contain only composed image examples. In 3.8, we discuss the creation of a dataset to fill this purpose. However, such a requirement could create a substantial hurdle for later work in x-ray image composition. To address this, we extend our network to facilitate unpaired training by generating individual threat and baggage scans from unpaired examples. We can separate the threat item from the bag using manual annotations, providing the threat scan but leaving gaps in the baggage. Reformatting our problem, we have created paired data if we can fill these gaps. For this task, as suggested by [4], we apply an image inpainting network.

Inpainting networks are used widely to repair images. They can be applied to many tasks, such as removing text, noise or scratches from images. We are interested in reconstructing images after object removal. As proposed in [4], a self-supervised conditional GAN structured identically to our colour consistency network can be used to learn the image context and produce a plausible hypothesis for the missing region. To train our inpainting network, we create threat-shaped holes in negative baggage scans. With these images, we can evaluate the realism of an inpainted image through comparison with original scans. Using an example threat item, we could make these holes randomly. However, with this strategy, we may train our network to inpaint areas which are not representative of those required to train composition. Abstracting, we can frame realistic threat holes prediction in negative baggage as the prediction of realistic locations for threat item insertion; we can use a pretrained geometric consistency module to create realistic holes.

In summary, to allow unpaired composition training, we require a spatial transformer network and an image inpainting network. First, we train the spatial transformer to predict threat object locations within baggage. This model can then create paired imagery to train the inpainting network by creating realistic threat-shaped holes in baggage imagery. Finally, with both of these pretrained components, we can extract paired images from unpaired examples to train our end-to-end image composition network. A closer view of the training procedure can be seen in figure 6.

3.6 Our Architecture

Combining the geometric and colour consistency networks dicsused previously, we produce the composition network shown in figure 6. A key addition in our end-to-end architecture is the introduction of cycle consistency. As proposed by [4], for higher quality composites we introduce a supervisory signal ensuring that objects can be decomposed realistically. This creates a stronger relationship between the threat items and baggage, reducing mode collapse and improving the realism of composites.

To enforce cycle consistency, we create decomposition modules for geometric and colour consistency. In our geometric composition network, a spatial transformer network returns a translated object to its original geometry. Again, we use a geometric consistency loss, described in 3.3, comparing the decomposition output to the input of the composition translation. Likewise, for colour consistency, we use a decomposiiton module structured similarly to the conditional GAN described in 3.4. However, inversely to the composition module, we take the composed image as auxiliary information and predict the decomposed baggage-threat pair. Considering this, our end-to-end composition network G, with geometric modules G^c_{stn}, G^d_{stn}, and colour consistency generator-decoder pairs G^c_{col}, D^c_{col}, G^d_{col}, and D^d_{col} we obtain the overall objective function,

$$\begin{aligned}\mathcal{L}(G) = \; & \lambda_1[\mathcal{L}_{\text{L1}}(G^c_{\text{col}}) + \mathcal{L}_{\text{L1}}(G^d_{\text{col}})] \\ + \; & \lambda_2[\mathcal{L}_{\text{geometric}}(G^c_{\text{stn}}) + \mathcal{L}_{\text{geometric}}(G^d_{\text{stn}})] \\ + \; & \lambda_3[\mathcal{L}_{\text{cGAN}}(G^c_{\text{col}}, D^c_{\text{col}}) + \mathcal{L}_{\text{cGAN}}(G^c_{\text{col}}, D^c_{\text{col}})]\end{aligned} \tag{7}$$

Using this architecture, we learn realistic spatial locations for threat objects and then synthesise insertion. Beyond novel experimentation with deep-learning-based composition architectures for x-ray images, we have implemented various segmentation loss functions to supervise the geometric consistency to produce an architecture adapted to our task. Additionally, as proposed by [4], we have extended the architecture to consider the unpaired case using a self-supervised image inpainting network. Overall, our finetuned end-to-end composition network learns to create realistic composition to be used in a variety of applications.

3.7 Object Detection Networks

Our motivation to include object detection techniques within this work is two-fold. Firstly, it allows us to generate a quantitative evaluation of our work. Using the change in detection performance, we can assess the realism of our composites. Additionally, such experimentation provides a novel investigation into the use of deep-learning-based composites as training data for object detection networks. Due to the lack of x-ray imagery, these tools could prove significant in producing accurate threat item detection networks for general use in the security community. When selecting architectures, we consider the performance in prior work on x-ray datasets and the advancements made on natural imagery datasets. We used two region-based CNN detectors: Cascade R-CNN and FSAF. These build on previous work; Cascade R-CNN is used in [70], the current state-of-the-art on SIXRay and OPIXRay, and builds on previous Faster R-CNN experimentation [7]. FSAF extends RetinaNet architectures used for x-ray object detection in [22]. Finally, as FSAF and Cascade R-CNN are from different architectural families, we can evidence conclusions generalisable across detection methods; this is key to understanding and evaluating the quantitative performance of our composites.

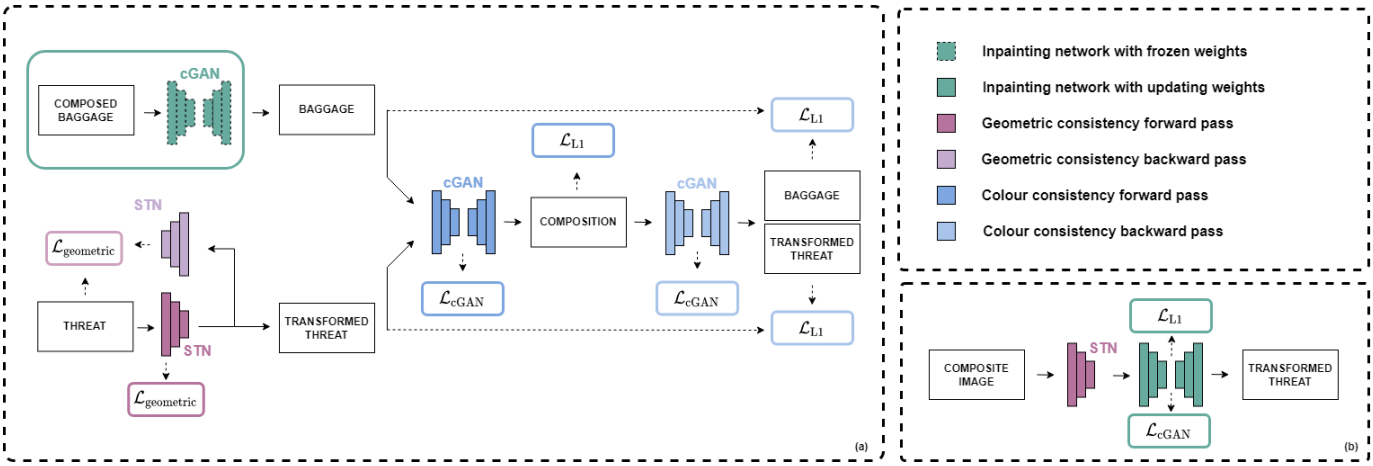

Figure 6: Composition network architecture. Section (a) shows the end-to-end training architecture. Section (b) shows the inpainting training architecure.

Both Cascade R-CNN and FSAF are designed using a ResNet-101 backbone. As discussed previously, ResNet architectures apply skip-connections to reduce information degradation allowing deeper networks to extract pertinent information and learn useful representations. After passing through an initial 7×7 convolutional filter, we build our backbone network using a series of ResNet blocks containing a stack of three convolutional layers. In contrast to our cGAN generator, ResNet block convolutional layers have 1×1, 3×3 and 1×1 kernel sizes, respectively. Again, the skip connection joins the layers immediately surrounding each block. Our backbone contains 33 ResNet blocks contributing 99 of a total of 101 convolutional layers in the network. By feeding images through our backbone, we learn a representation from which both Cascade R-CNN and FSAF make bounding box predictions of threat item locations.

After passing the image through the backbone, both architectures use a Feature Pyramid network. The FPN creates an effective method of detecting objects on multiple scales; it contains a bottom-up and top-down pathway, in our implementation both have a five-layer pyramidal structure. First, the bottom-down network uses convolutional blocks to downsample the feature map by a factor of two iteratively. Despite decreasing the spatial dimension, this extracts features of increasing semantic value. Following the bottom-up network, we feed to the top-down layers through a 1×1 filter, reducing to a 256 channel feature map. Our top-down network iteratively upsamples the representation to distribute semantic information over multiple scales and combines with channel regularised bottom-down layers at each step via pixel-wise addition to retain spatial details. Through this process, the FPN extracts a set of multi-scale features. Therefore, after passing every upsample through a 3×3 convolutional filter to reduce the aliasing effect, each layer is fed to bounding box predicition subnetworks.

Figure 7: Example object detection predictions.

3.7.1 *Cascade R-CNN*

As with all region-based CNNs, Cascade R-CNN passes the learned multi-scale representation to a Region Proposal Network (RPN). Firstly, to clarify notation, we define anchors as points in a feature map about which we predict bounding boxes, also known as anchor boxes. Cascade R-CNN is an anchor-based method; bounding boxes are predicted solely using anchor boxes. Taking each pyramid level as input, the RPN is applied convolutionally, predicting bounding boxes over eight scales and three aspect ratios for each anchor using a 3×3 convolutional filter followed by two parallel 1×1 filters. The first filter predicts two probabilities, representing the odds that each bounding box contains an object or a background. These values are used later in the network for box refinement and suppression. Simultaneously, the second 1×1 layer predicts offsets for the location and dimensions of bounding boxes. As all bounding boxes are initialised in standard locations, scales, and aspect ratios these offsets are crucial to finetuning box predictions.

After extracting region proposals from the final feature map produced by the backbone with an RoI Align module, Cascade R-CNN attempts to classify and refine bounding box predictions using three sequential detection heads. Each detection head contains a classifier and regressor subnetwork, which accept a flattened representation learned by the aforementioned architecture describing the proposal area. The classifier reduces the flattened proposal to a 3×1 vector encoding a prediction as either knife, gun or background. Similarly, the bounding box regressor reduces to a 4×1 vector which encodes bounding box offsets, allowing further adjustments to the proposal. After these predictions are made, we feed the updated proposals to the next detection head. This sequential process allows iterative improvement of bounding box localisation, with each step at least as good as the previous. Following the final box predictions, we apply non-maximum suppression (NMS). By predicting many bounding boxes for each anchor, we often find that several proposals attempt to bound the same object. Non-maximum suppression checks for such redundancy between every bounding box in a particular class. If the Intersection over Union (IoU) between any two bounding boxes of the same class is above 0.7, NMS discards the box with a lower confidence score. Here, the confidence score is calculated

as a product of the objectness score produced in the RPN and the conditional class probability produced from the classification subnetwork. Following NMS, we output our final object detection predictions.

3.7.2 FSAF

In the FSAF network, we introduce hybrid anchor-based and anchor-free prediction. In anchor-based networks, we generate bounding box predictions about arbitrarily defined anchors generally given by the highest FPN level. This dependency on a hand-crafted heuristic can be challenging to optimise. If we do not choose our anchors correctly, it may lead to suboptimal performance. FSAF introduces an anchor-free branch parallel to a RetinaNet anchor-based branch to avoid this.

As with Cascade R-CNN, the FSAF architecture uses a ResNet-101 backbone followed by a five-layer feature pyramid network. Each feature layer is passed through a shared network head consisting of four 3×3 convolutional layers. From here, we split into our two branches - providing the result of the network head to both. Each branch contains two small fully-convolutional subnetworks for classification and regression. Following RetinaNet, our anchor-based classification subnetwork predicts the probability of each object class at each spatial location for each anchor; for a K class task with A anchors, the subnetwork outputs a KA-dimensional classification prediction. Similarly, for each anchor, the regression subnetwork outputs a 4-dimensional feature map predicting the distance from each spatial location. In contrast, the anchor-free classification subnetwork produces a K-dimensional map estimating the class conditional probability of each spatial location containing an object. In addition, the regression subnetwork predicts the distance from each location to a bounding box. We gather bounding box proposals from both branches during prediction, applying a non-maximum suppression to reduce proposal redundancy. By generating bounding box proposals from both the anchor-based and anchor-free branches, the FSAF network is trained in a multi-task style.

3.8 Training and Evaluation Datasets

As discussed previously, we experimented with both paired and unpaired data to train our composition model. For unpaired data, we used dbf3 [7]. This dataset was chosen for two key reasons. Firstly, segmentation information detailing the location of threat items is available. These annotations are crucial to extending into the unpaired domain; with segmentations, we can gather the threat item and use an image inpainting network to generate a negative bag. Secondly, using dbf3 allows for comparative evaluation against similar work [7]. To our knowledge, dbf3 is the only dataset on which similar x-ray image composites have been investigated with object detection models, in which [7] achieved an mAP of 81%. Motivated by a lack of firearm part imagery, we restricted training to firearm and knife classes. Within dbf3, there are only 1,203 firearm parts images creating a class imbalance in the dataset. Significantly lower information within certain classes can cause poor predictive performance. As such, to best understand the abilities and limitations of our network, we decided to leave such extensions as future work.

To generate our paired imagery, we scanned images using a Gilardoni FEP ME 640 AMX threat detection scanner. We scanned six distinct bags with five firearms and five knives; furthermore, we generated numerous scans with varying threat poses for each threat-bag combination to produce a diverse dataset suitable for training. After scanning, we manually annotated each image to obtain threat image transposition information. Again, we aimed to produce a balanced dataset. As shown in table 1, we generated 899 paired images, 385 firearm and 514 knife. We do not need to estimate baggage pairs; this produces several benefits. First, as we do not need to estimate the appearance of the negative bag, we reduce the performance bottleneck produced by the inpainting network. On unpaired imagery, the realism of the resultant composite is significantly influenced by the realism of the inpainting. If our inpainting network performs poorly, we may not be able to produce a generalisable model as our composition network could exploit imperfections in the inpainting to overfit on a training set. Moreover, using paired imagery removes the need to pre-train an STN and image inpainting network. This architectural simplification reduces the training time and the size of our model.

When considering how to create a quantitative evaluation of our composite imagery, we decided on five key categories. To ensure clarity in the number of images present, the proportions in hybrid dataset names are given relative to the total number of images in the dbf3 dataset (e.g 33% synthetic refers to the use of 1,477 synthetic images, regardless of the total images in the dataset):

(i) *Baseline validation (dbf3)*: to produce a baseline to which we can compare the use of our synthetic images, we used the dbf3 dataset.
(ii) *Hybrid (100:33)*: a hybrid dataset containing 4,433 images from dbf3 (100%) supplimented with an additional 1,477 synthetic images (33%).
(iii) *Hybrid (50:50)*: a hybrid dataset containing 2,217 images from dbf3 (50%) and 2,216 synthetic images (50%).
(iv) *Composite (4,433)*: a fully composite dataset containing 4,433 images, identical to the amount in dbf3.
(v) *Composite (20,000)*: a fully composite dataset containing 20,000.

Evaluating the success of our network using both hybrid and composite imagery allows for an extensive investigation into the possible uses of our network. The hybrid datasets were chosen to investigate the use of composite imagery as supplementation to real images. Similarly to data augmentation techniques, such datasets increase the variety of construction and pose for data when training object detection networks. As such, we created two hybrid datasets. The hybrid (50:50) dataset evaluates composite imagery in place of real data, whereas the hybrid (100:33) evaluates its use in addition to real data.

Similarly, we explore the use of training exclusively using composite data. Our composite datasets allow for direct comparisons to baseline results and create an understanding of the overall efficacy of our approach for object detection networks. To evaluate this, we use two composite image datasets. The first, composition (4,433), is sized identically to dbf3, qualifying a comparative discussion against real data. The second draws on a significant strength of image composition networks. Unlike traditional x-ray dataset generation involving time-consuming manual scanning and annotation, we can generate an arbitrary amount of data with minimal effort. To investigate the effect of increased synthetic training data, we generated 20,000 composite images. Increasing the training data by over four times, composition (20,000) representatively shows the performance of such techniques; generating additional images would likely show diminishing returns.

Table 1: Class-wise breakdown of the datasets used in our solution.

Dataset		Class		
Type	Name	Firearm	Knife	Total
Composition Training	dbf3 [7]	3,161	3,154	6,345
	paired	385	514	899
Detection Training	$\text{dbf3}_{\text{train}}$ [7]	2,209	2,224	4,433
	$\text{hybrid}_{100:33}$	2,948	2,962	5,910
	$\text{hybrid}_{50:50}$	2,217	2,216	4,433
	$\text{comp}_{4,433}$	2,217	2,216	4,433
	$\text{comp}_{20,000}$	10,000	10,000	20,000
Evaluation	dbf3_{val} [7]	952	960	1912

4 Results

As with many deep neural networks, the computational demands of training our model exceeded standard computer hardware capabilities. Due to their ability to perform parallel computing, GPUs are often used. We trained our model using the Durham NVIDIA CUDA Centre (NCC) GPU system. We detail the initialisation and hyperparameters of our models independently.

Motivated by the use of ReLU activation functions throughout our composition network, to prevent vanishing or exploding gradients, we used Kaiming weight initialisation. Tailored for deep neural networks using asymmetric functions, Kaiming initialisation accounts for non-linearity in ReLU activation functions to provide training stability. We used an initial learning rate of 2×10^{-5} with a lambda learning-rate scheduler during training. Our end-to-end model was trained for 1000 epochs, with linear learning-rate decay after the 750$^{\text{th}}$ epoch. We pre-trained an STN and inpainting network for 500 epochs at the same learning rate before the end-to-end model to generate paired examples when using unpaired data. For validation, we checkpoint at regular intervals and examine performance of baseline detector on 100 images produced using the checkpointed model. This allows us to find the model able to generate the most realistic imagery. An Adam optimiser was used in backpropagation with beta values of 0.5 and 0.999 to provide an efficient and effective optimisation strategy.

As shown in [2, 70], pre-trained weight initialisation outperforms random, Xavier, and Kaiming initialisations for threat object detection; our FSAF and Cascade R-CNN detectors were initialised with weights provided by MMDetection [12] after training on the COCO dataset [39]. The models are trained for 30 epochs with a batch size of 50 in all experiments. Our models trained with a standard learning rate of 2×10^{-2} using a 500 iteration warmup and a warmup ratio of 1×10^{-3}. Backpropagation optimisation is performed via an SGD optimiser with a momentum of 0.9 and weight decay of 1×10^{-4}. We validate against dbf3_{val} and checkpoint at every epoch to find the highest performing model. Finally, as suggested in [10], our Cascade R-CNN model uses detector head IoU thresholds $U = \{0.5, 0.6, 0.7\}$.

Due to time and memory limitations, we trained our composition model on imagery resized to 128×128 pixels; this restriction allowed for a much more varied and comprehensive set of investigations. However, it is essential to note that decreasing image resolution affects image quality. We perform all object detection and comparisons on resized imagery to provide a fair evaluation of our composite imagery.

To compare with previous papers and assess the quality of our model we will evaluate using COCO metrics [39]. Considering each class independently, a bounding box is said to be true positive (TP) if it has an IoU > 0.5 with an identically classified ground truth annotation. Otherwise, the predicition is said to be false positive (FP). We measure the average precision of a class (AP) and mean average precision (mAP) over the entire test set using,

$$\text{AP} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (8) \qquad \text{mAP} = \frac{1}{K} \sum_{i=1}^{K} \text{AP}_i \quad (9)$$

where K represents the number of classes and AP_i the average precision of class i. We will also undertake a qualitative evaluation of our imagery. We visually compare images to understand the differences between training methods and provide additional context for our object detection results, as well as assessing its applicability in threat image projection.

4.1 STN Loss Function

Motivated by the poor performance of L1 loss, we experimented with the use of three new loss functions. To improve the transformation of threat imagery for insertion in composite imagery, we tested the use of Dice loss, Tversky loss and Focal Tversky loss. Unlike L1 loss, each proposed loss function evaluates the success of translation invariant to object scale. Regardless of domain, this is a vital attribute. However, it is especially crucial in x-ray imagery which can contain threat items with a diverse range of scale and rotation. All results shown in table 2 were obtained by training the composite network on unpaired imagery. Therefore, we used our geometric loss to supervise both the spatial transformer pre-training for image inpainting and end-to-end training.

To provide a benchmark, we first discuss the performance of L1 loss. As suspected, the network produces very poor quality composite imagery. Table 2 shows that neither object detection network made any correct bounding box predictions, producing an mAP of 0%. As shown in figure 8, due to an imbalance in the penalty produced by L1 loss relative to object size, the STN transforms threat items to occupy just a few pixels. Such transformations create two key problems. Firstly, training on L1 composite imagery does not allow models to learn detection on a larger, more realistic scale. The detector cannot learn to extract generalised semantic information describing threat items. Secondly, as the predicted bounding boxes become very small, they are unlikely to reach the IoU > 0.5 threshold even if positioned correctly, leading to no true positive bounding box positions.

Table 2: Loss function test results.

Detector	Loss Function	Dataset (mAP)			
		$\text{h}_{100:33}$	$\text{h}_{50:50}$	$\text{c}_{4,433}$	$\text{c}_{20,000}$
Cascade R-CNN	Dice	0.758	0.712	0.094	0.022
	Tversky	**0.760**	0.714	0.204	0.133
	Focal Tversky	0.754	0.704	**0.280**	**0.226**
	L1	0.687	**0.732**	0.000	0.000
FSAF	Dice	0.681	**0.652**	0.021	0.031
	Tversky	0.680	0.649	0.187	0.172
	Focal Tversky	0.686	0.616	**0.208**	**0.207**
	L1	**0.690**	0.249	0.000	0.000

Including the proposed geometric loss functions vastly improves the performance of detectors trained on composite imagery; detection networks trained with Dice, Tversky and Focal Tversky imagery outperform those trained on L1 loss composites. In our

tests, Focal Tversky produced the best composite imagery with an mAP of 28.0% and 20.8% on $c_{4,433}$ achieved by a Cascade R-CNN and FSAF detector, respectively. This greatly improves upon L1 composite imagery, but also outperforms Tversky and Dice. These results imply that, out of the loss functions we tested, the Focal Tversky loss best supervises the STN to learn the realistic geometric warping. It follows that the composites generated are closest to the target domain. Additionally, as Tversky imagery outperformed Dice imagery, each of the hyperparameters introduced to the loss function improve the performance of the model. We used the Focal Tversky loss in all further testing.

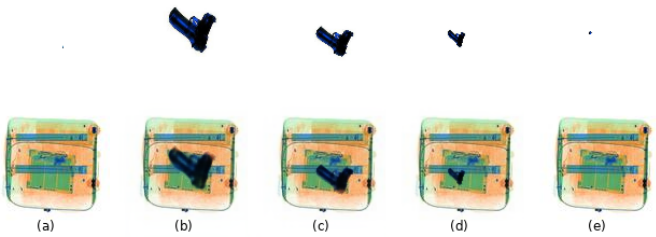

Figure 8: Example composite produced with geometric loss (a) L1, (b) Dice, (c) Tversky, (d) Focal Tversky and (e) Focal Tversky with data augmentation.

Evaluating the qualitative performance of models supports these results. As can be seen in figure 8, given the same input baggage-threat pair, the spatial transformer network supervised by Focal Tversky loss generates the most realistic transformation and subsequent composite. The scale of the transformed threat image produced by both the Dice loss transformer and Tversky loss transformer does not match the scale of the bag. As discussed previously, the L1 loss transformer reduces the threat item to just a few pixels. Consequently, the generated composite image appears identical to a negative baggage scan.

It is important to note how our results imply that hybrid datasets are an unreliable indicator of composite imagery realism on their own. As the detector can rely heavily on the real examples during training, the quality of composites is less influential to hybrid performance. Interestingly, we see a near inverse relationship between composite and hybrid performance. Composite imagery that performs poorly on its own seems to facilitate higher accuracy hybrid detectors across Cascade R-CNN and FSAF. For example, on $h_{50:50}$ data, L1 imagery produces the most accurate Cascade R-CNN model, and Dice imagery generates the best FSAF model, outperforming other models by 1.8% and 0.5%, respectively. We suspect this occurs as the detector learns to distinguish real and composite images during training. Therefore, during testing, the performance of our model is comparable to that of training without composites. As it better reflects the performance of models, we use composite imagery as the primary quantitative metric to understand the realism of generated images.

4.2 Batch Size

During loss function testing, time constraints became a significant bottleneck to the amount of experimentation we were able to undertake. Azadi et al. [4] propose a batch size of 16; however, as shown in table 3 this takes over four days to train. To reduce the training time, we investigated models with varying batch sizes. The batch size determines the number of predicted images evaluated per training step. Therefore, larger batch sizes require fewer training steps to iterate over the same dataset. Intuitively, increasing the batch size allows for greater parallelisation of tasks. Following the standard power of two batch sizes, we explored models trained using batches of 32 and 64 images. Larger sizes were not tested as we found the training time between the two tests to be similar. Therefore, increasing the batch size further is unlikely to provide any training time benefit. We benchmarked the results against our best performing model with a batch size of 16, produced using the Focal Tversky loss.

Table 3: Training times for different batch sizes.

Batch Size	16	32	64
Time (hrs)	142.7	71.9	68.0

As suspected, increasing the batch size reduces the training time significantly, to approximately half, with a batch size of either 32 or 64. However, the quality of composite imagery is reduced. As shown in table 4, the performance of object detection networks trained on larger batch synthetic imagery is considerably lower than our benchmark. Across both detectors, mean average precision did not exceed 1% for either $c_{4,433}$ or $c_{20,000}$. Despite this, hybrid performance increased with both batch sizes. Again, this suggests hybrid results are an unreliable metric for composite realism if the quality of imagery is too low.

Table 4: Batch size test results.

Detector	Batch Size	Dataset (mAP)			
		$h_{100:33}$	$h_{50:50}$	$c_{4,433}$	$c_{20,000}$
Cascade R-CNN	16	0.754	0.704	**0.280**	**0.226**
	32	0.758	0.702	0.001	0.001
	64	**0.762**	**0.708**	0.005	0.004
FSAF	16	**0.686**	0.616	**0.208**	**0.207**
	32	0.676	**0.634**	0.005	0.001
	64	0.677	0.632	0.004	0.005

The performance decrease of our model on larger batch sizes is not unique; increasing the batch size has been shown to impact the performance of models in various tasks negatively [26, 73]. Commonly, increasing the batch size impacts the ability of a deep-learning-based model to learn a generalisable relationship between inputs and outputs. As fewer training steps are made when using larger batch sizes, the model can perform fewer parameter updates to adjust for inaccurate predictions and learn realistic composition. The training time decrease afforded for our composition model does not justify such a steep performance decrease, so we use a batch size of 16 for all subsequent results.

4.3 Paired Data and Data Augmentation

After tuning hyperparameters and experimenting with loss functions, we investigated the use of paired data and data augmentation techniques. Data augmentation can significantly increase the variance in training data, artificially increasing the size of the training set. Therefore, we can reduce overfitting to produce a more generalisable model. As shown by [70], data augmentation techniques can increase the performance of deep learning models within the x-ray domain. Webb et al. improved upon a baseline object detection performance by 0.9% using flipping, 3.1% using cropping, and 0.9% using rotation on OPIXRay with a Cascade R-CNN detector [70]. We implemented these three techniques to

understand the effect of data augmentation on our model. In our data augmentation training scheme, there is a 50% chance that each loaded image is flipped (horizontally or vertically), cropped by up to 60% its original size or rotated by some multiple of 90 degrees. The probability of each augmentation is independent, and several augmentation techniques may occur on the same image. To provide an in-depth understanding of the effect of differing training data for our composition network, these were applied to the unpaired dbf3 data and the paired dataset produced for this work.

As displayed in table 5, our model failed to produce realistic composites in each of the tests. Detection networks utilising composites generated after training with data augmentation, paired imagery, or a combination of the two, fail to correctly predict any bounding boxes during testing. First, we discuss the paired training results. As can be seen in figure 11, models trained on our paired dataset were unable to create composites with any sense of realism. A qualitative comparison between our best unpaired and paired images highlights why object detection models performed so poorly on paired $c_{4,433}$ and $c_{20,000}$ datasets. However, these results are not caused by an inherent architectural flaw limiting performance on paired imagery. Instead, they are a joint consequence of shortcomings in the paired dataset and our choice of loss function.

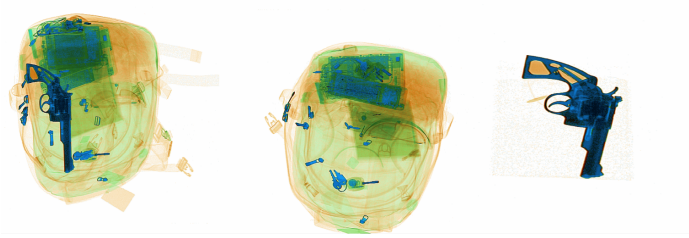

Figure 9: Example paired training image.

To create paired data, we must match baggage containing a threat item with a negative bag. These two bags must have an identical pose and location in their images. A precise match is very tough to achieve; this is exemplified in figure 9, slight variation in the position and pose of the bag is hard to avoid. As we use an L1 loss, the network is penalised for these variations. As a result, our model begins to predict the random dataset variation, causing the composites to lose any sense of realism.

Table 5: Data augmentation and paired training results.

Detector	Model Strategy	Dataset (mAP)			
		$h_{100:33}$	$h_{50:50}$	$c_{4,433}$	$c_{20,000}$
Cascade R-CNN	Unpaired	0.754	0.704	**0.280**	**0.226**
	Unpaired Aug	0.750	0.710	0.000	0.000
	Paired	**0.767**	0.716	0.000	0.000
	Paired Aug	**0.767**	**0.734**	0.000	0.000
FSAF	Unpaired	0.686	0.616	**0.208**	**0.207**
	Unpaired Aug	0.679	0.606	0.000	0.000
	Paired	**0.710**	**0.678**	0.000	0.000
	Paired Aug	0.709	0.671	0.000	0.000

Data augmentation caused underfitting to occur in geometric and colour subnetworks on unpaired imagery. As shown in figure 8, transformations appear massively scaled-down. Moreover, comparing augmented unpaired composites with the baseline unpaired composites in figure 11 we can see that imagery became blurred around the inserted threat item. This is reflected in table 5; the reduced performance of our data augmentation network is clear. Although it is difficult to measure the effect of data augmentation on paired imagery, contrary to unpaired training, it appears as though it may have slightly improved network performance. The model trained with data augmentation appears to have captured a greater amount of fine image details, composites resemble a smoothed version of the negative bag. However, this is inconsequential to the performance of object detection networks on the imagery.

4.4 State-Of-The-Art Comparison

Finally, we provide a comparison with the state-of-the-art. As the best model produced through our experimentation, we evaluate the composite imagery generated by our unpaired Focal Tversky model against the TIP imagery produced in [7]. To benchmark the performance of both methods, we show baseline results produced by training Cascade R-CNN and FSAF detectors on $dbf3_{train}$. For each dataset, we examine the mean average precision and class-wise precision to evaluate the strengths and limitations of different approaches. As we restricted our model to 128 $\times$ 128 pixel images, to provide a fair comparison, the results shown in table 6 are performed on similarly resized dbf3 and TIP imagery (provided by [7]). It is important to note that we were only able to obtain 12,000 TIP images, so we replaced the $c_{20,000}$ dataset with $c_{12,000}$ containing 12,000 composite images. As the number of images in $c_{20,000}$ was chosen to be arbitrarily large, this should not cause any significant performance difference, but it should be taken into account.

As shown in table 6, the best result by both composition strategies was on $h_{100:33}$. On this dataset, our model achieved an mAP of 75.4% and 68.6% using a Cascade R-CNN and FSAF detector, respectively. In comparison, the best result achieved using TIP was 76.8% using Cascade R-CNN and 70.9% using FSAF. On average, models performed 0.7% better using TIP images in $h_{100:33}$ than on our imagery. The performance of detectors on these hybrid datasets is similar to the baseline results, which achieve an mAP of 76.6% using Cascade R-CNN and 72.2% using FSAF. Although a slight improvement is shown when training using TIP imagery on $h_{100:33}$, none of the hybrid datasets were able to improve upon the baseline result significantly.

Figure 10: Example failure cases.

In contrast to hybrid dataset results, TIP composite datasets notably outperformed ours. This is exemplified when considering

Table 6: State-of-the-art comparison results. The left table shows results produced with a Cascade R-CNN detector and the right table shows results produced with an FSAF detector. Bold results denote the best performance on a dataset and underlined results denote the best performance in a metric.

Method	Dataset	Precision		
		Firearm (AP)	Knife (AP)	mAP
Ours (Focal Tversky Unpaired)	$\text{hybrid}_{100:33}$	**0.978**	0.529	0.754
	$\text{hybrid}_{50:50}$	**0.973**	0.435	0.704
	$\text{comp}_{4,433}$	0.536	**0.024**	0.280
	$\text{comp}_{20,000}$	0.447	**0.005**	0.226
TIP	$\text{hybrid}_{100:33}$	**0.978**	**0.557**	**0.768**
	$\text{hybrid}_{50:50}$	0.966	**0.506**	**0.736**
	$\text{comp}_{4,433}$	**0.591**	0.000	**0.296**
	$\text{comp}_{20,000}$	**0.705**	0.000	**0.352**
Baseline	$\text{dbf3}_{\text{train}}$	**0.978**	0.556	0.766

Method	Dataset	Precision		
		Firearm (AP)	Knife (AP)	mAP
Ours (Focal Tversky Unpaired)	$\text{hybrid}_{100:33}$	0.966	0.405	0.686
	$\text{hybrid}_{50:50}$	0.947	0.284	0.616
	$\text{comp}_{4,433}$	0.406	**0.010**	0.208
	$\text{comp}_{20,000}$	0.404	**0.010**	0.207
TIP	$\text{hybrid}_{100:33}$	**0.970**	**0.448**	**0.709**
	$\text{hybrid}_{50:50}$	**0.950**	**0.378**	**0.664**
	$\text{comp}_{4,433}$	**0.537**	0.000	**0.268**
	$\text{comp}_{12,000}$	**0.562**	0.000	**0.281**
Baseline	$\text{dbf3}_{\text{train}}$	**0.980**	**0.465**	**0.722**

the change in performance of detectors between the baseline and larger composite image datasets. Using TIP images, composite performance improves on larger datasets by 5.6% with a Cascade R-CNN detector and 1.3% with an FSAF detector. Whereas, detection precision decreased by 5.4% using Cascade R-CNN and 0.1% using FSAF between our composite datasets. Regardless, neither method produces a composite dataset able to outperform real data. Using a Cascade R-CNN detector, the baseline result beats our composite dataset by 48.6% and TIP composites by 42.4%. These results suggest that the composite imagery produced by [7] is more realistic than composites produced using our model, but they are still an insufficient substitute for real data.

Detection models trained on data produced by either composition technique show similar class-wise precision trends; performance in the firearm class is much better than in the knife class. This pattern is also reflected by the baseline result, although to a lesser extent. These results suggest that the knife class is significantly more challenging. This can be explained by the greater variation in the shape and appearance of knives compared to firearms. Within the knife class, our model outperforms TIP. A Cascade R-CNN detector achieves class-wise precision of 2.4% on our $c_{4,433}$ dataset; however, no correct knife predictions were made after training on TIP data. Although this is a small sample, it highlights the ability of our model to outperform random insertion in complex domains.

Figure 11 shows a qualitative comparison of the two methods. Generally, our predicted insertion locations better match the background context than TIP imagery. In particular, note the example firearm orthogonal to the detector in figure 11. Using TIP, the firearm is inserted randomly and produces an unrealistic composite where the firearm appears central in the bag. In contrast, our model predicts the scale and position of the orthogonal firearm to create a more realistic composite where the firearm appears on the side of the bag.

A key difference between TIP imagery and our imagery is the severity failure cases. Using TIP, the least realistic images may contain unnatural insertion locations. In contrast, figure 10 shows some common failure cases for our model. It follows that, the failure cases produced by our model have an outsized impact on object detection performance. Despite an ability to generate realistic composite imagery, these failure cases substatially decrease the usability of our imagery. Our results show the potential of deep-learning-based x-ray composition. However, it is clear that further work must be undertaken to stabilise the network and improve the reliability of generated imagery.

5 Evaluation

In this section, we will evaluate the strengths and limitations of our final solution within the context of our research question: *Can contemporary advances in deep-learning-based composition networks be successfully applied to the domain of x-ray imagery?*

5.1 Solution Strengths

We consider the extensive testing performed in this work a notable strength. To our knowledge, this is the first work to investigate the use of deep-learning-based image composition within the x-ray domain. In this novel experimentation, we provide a detailed qualitative and quantitative evaluation of the performance of such methods. Although we did not generate composite imagery able to improve threat object detection accuracy on x-ray imagery, we have demonstrated the significant promise of deep-learning-based models in this domain. Through a comprehensive investigation into the use of paired and unpaired imagery, varying batch sizes, differing loss functions, and data augmentation techniques, we created a model able to generate realistic composite imagery. Moreover, in many cases, our model can successfully assess the visual context of baggage-threat pairs to improve upon the realism of traditional TIP composites. Though further work should be undertaken to finetune performance, this work suggests an affirmative answer to the research question.

Another considerable strength of our solution is the novel experimentation with the use of Dice, Tversky and Focal Tversky losses to evaluate the performance of spatial transformer networks. Both qualitative and quantitative evaluations show that the L1 loss proposed by [4] performs poorly. In adapting the model to our task, investigations using the aforementioned loss functions outperformed this baseline significantly. Beyond composition, this experimentation informs research across a broad range of domains. Our results motivate further exploration into the use of these loss functions in spatial transformers.

5.2 Solution Limitations

A clear, fundamental limitation of our solution was the inability to improve the performance of object detection models on x-ray imagery. The baseline $\text{dbf3}_{\text{train}}$ dataset outperformed our best dataset by 1.2% on a Cascade R-CNN detector and 3.6% on an FSAF detector. To this point, our solution did not improve upon the TIP imagery generated in [7]. A key reason for this is the instability of our model. Not only does our solution produce a

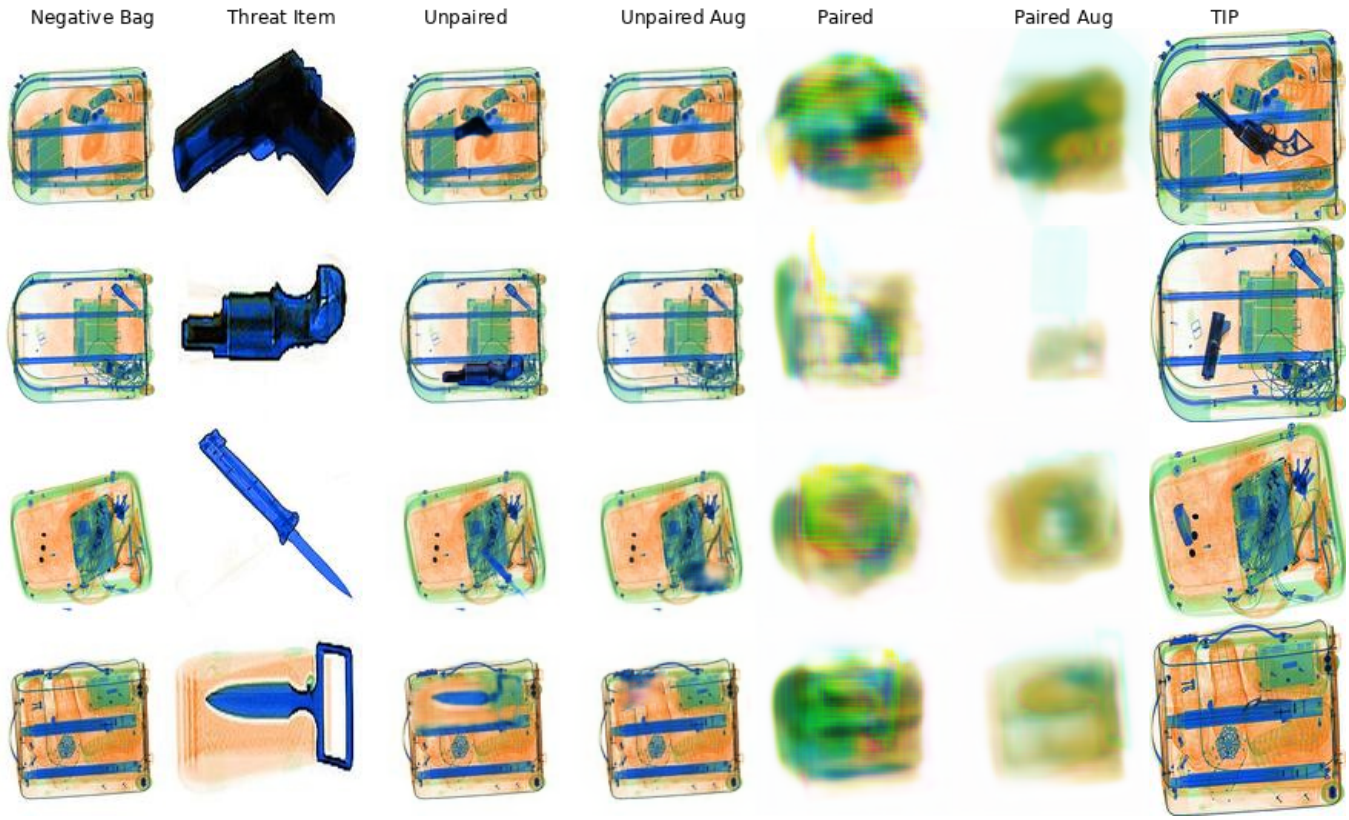

Figure 11: Example images produced using different composition techniques. All models except TIP were given the same input baggage-threat pair.

large number of failure cases, but it is challenging to optimise. Our testing over different batch sizes and data augmentation techniques evidences how small changes to training techniques have an outsized impact on composite realism. Additionally, our model takes a considerable amount of time to train. In conjunction with the difficulty of optimisation, these time limitations dramatically reduced the amount of testing we were able to undertake.

The quality of our paired dataset limited our solution significantly. Combined with L1 loss in the colour composition network, the difficulty of matching paired images caused paired dataset tests to fail. Object detection models trained on composite paired datasets could not accurately predict the bounding boxes of any imagery in the test dataset. As such, the bottleneck created by the inpainting network used on unpaired imagery could not be sufficiently investigated. Similarly, motivated by to time and memory constraints, the restriction of investigations to 128×128 pixel images provided a notable limitation. Downsampling and resizing imagery can significantly affect the quality of images and does not effectively represent the possible uses of our models in practice. For these reasons, we cannot definitively state that we have explored the full capability of our model.

5.3 Future Work

Extensions to our research to improve the reliability and realism of composites can be split into two categories: architectural changes and dataset changes. For the former, we propose two key ideas. Highlighted by underfitting with data augmentation; further research could investigate the use of deeper, more complex models. This could allow the model to extract higher quality semantic information, facilitating a more generalisable relationship between inputs and outputs. Secondly, to lessen failure cases, we could attempt to reduce the difficulty of image insertion. For example, similarly to [74, 75], we could provide the colour consistency network with transformed threat images superimposed onto negative baggage rather than a baggage-threat image pair. As the colour consistency network does not need to learn how to insert a threat item and can focus on the visual appearance after insertion, this change eliminates a crucial failure case exhibited by our network where transformed items do not appear in the composite.

For the latter extension, work focusing on exploring the capability of deep-learning-based x-ray image composition could focus on generating accurate paired imagery or investigate the use of larger image sizes. The motivation for paired imagery is clear; without estimating the appearance of the negative baggage, performance could improve considerably. Using a larger image size could provide similar benefits increasing the depth of the model. The additional information provided by the larger image size could allow the model more effectively extract semantic information connecting the baggage-threat pair with a realistic composite. Both techniques could provide notable performance increases without significant architectural change.

6 Conclusions

In this project, we produced a deep-learning-based image composition model capable of generating realistic synthetic x-ray imagery. To achieve this, we adapted the architecture proposed by [4]. Our model uses a spatial transformation network to consider the visual context of a baggage-threat pair to predict a

realistic insertion scale, rotation and location of the threat item, and a conditional GAN to improve the colour consistency of the generated composite. Building from the architecture proposed by Azadi et al. [4], we experimented with several geometric loss functions, data augmentation techniques and batch sizes on the unpaired dbf3 dataset [**dbf3**]. To compare methods, we undertook a qualitative evaluation comparing composite image realism and a quantitative evaluation considering the performance of object detection models trained on synthetic imagery. We concluded that the best model was produced using a Focal Tversky loss with a batch size of 16 without data augmentation. Furthermore, we investigated the use of paired imagery for our model by generating a paired dataset for this work. However, training on this imagery failed due to the difficulty of creating precisely matched paired examples.

Our solution was unable to outperform the baseline object detection results after training on real data from dbf3 or using TIP composition methods [7] due to a large number of failure cases. Despite this, our results provide an extensive novel investigation into the use of deep-learning-based image composition in the x-ray domain and clearly motivate continued research into the field. We believe that the progression of our project falls into two categories; architectural changes, such as the use of deeper spatial transformation or conditional GAN networks and dataset changes, such as the production of a diverse paired x-ray threat dataset.

REFERENCES

[1] N. Abraham and N.M. Khan. "A novel focal tversky loss function with improved attention u-net for lesion segmentation". In: *2019 IEEE 16th int. symposium on biomedical imaging (ISBI 2019)*. IEEE. 2019, pp. 683–687.

[2] S. Akçay et al. "Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery". In: *2016 IEEE Int. Conf. on Image Processing (ICIP)*. IEEE. 2016, pp. 1057–1061.

[3] S. Akçay et al. "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery". In: *IEEE transactions on information forensics and security* 13.9 (2018), pp. 2203–2215.

[4] S. Azadi et al. "Compositional gan: Learning image-conditional binary composition". In: *Int. Journal of Computer Vision* 128.10 (2020), pp. 2570–2585.

[5] M. Baştan, M.R. Yousefi, and T.M. Breuel. "Visual words on baggage X-ray images". In: *Int. conf. on computer analysis of images and patterns*. Springer. 2011, pp. 360–368.

[6] *BBC - Manchester Airport: Travellers miss flights amid chaos*. Online; accessed 12th April 2022. URL: https://www.bbc.co.uk/news/uk-england-manchester-60974266.

[7] N. Bhowmik et al. "The good, the bad and the ugly: Evaluating convolutional neural networks for prohibited item detection using real and synthetically composited X-ray imagery". In: (2019).

[8] P.J. Burt and E.H. Adelson. "A multiresolution spline with application to image mosaics". In: *ACM Transactions on Graphics (TOG)* 2.4 (1983), pp. 217–236.

[9] A. Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11 (2020).

[10] Z. Cai and N. Vasconcelos. "Cascade R-CNN: High Quality Object Detection and Instance Segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 43.5 (2019), pp. 1483–1498.

[11] B-C. Chen and A. Kae. "Toward realistic image compositing with adversarial learning". In: *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 2019, pp. 8415–8424.

[12] K. Chen et al. "MMDetection: Open MMLab Detection Toolbox and Benchmark". In: *arXiv preprint arXiv:1906.07155* (2019).

[13] L. Chen et al. "Quality-aware unpaired image-to-image translation". In: *IEEE Transactions on Multimedia* 21.10 (2019), pp. 2664–2674.

[14] European Commission. *Laying Down Detailed Measures for the Implementation of the Common Basic Standards on Aviation Security L 299/1*. 2015.

[15] U. Demir and G. Unal. "Patch-based image inpainting with generative adversarial networks". In: *arXiv preprint arXiv:1803.07422* (2018).

[16] J. Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conf. on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[17] N. Dvornik, J. Mairal, and C. Schmid. "Modeling visual context is key to augmenting object detection datasets". In: *Proc. European Conf. on Computer Vision (ECCV)*. 2018, pp. 364–380.

[18] D. Dwibedi, I. Misra, and M. Hebert. "Cut, paste and learn: Surprisingly easy synthesis for instance detection". In: *Proc. IEEE Int. Conf. on Computer Vision*. 2017, pp. 1301–1310.

[19] H-S. Fang et al. "Instaboost: Boosting instance segmentation via probability map guided copy-pasting". In: *Proc. IEEE/CVF Int. Conf. on Computer Vision*. 2019, pp. 682–691.

[20] L.A. Gatys, A.S. Ecker, and M. Bethge. "A neural algorithm of artistic style". In: (2015).

[21] L.A. Gatys et al. "Controlling perceptual factors in neural style transfer". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 2017, pp. 3985–3993.

[22] Y.F.A. Gaus et al. "Evaluating the transferability and adversarial discrimination of convolutional neural networks for threat object detection and classification within x-ray security imagery". In: *2019 18th IEEE Int. Conf. On Machine Learning And Applications (ICMLA)*. IEEE. 2019, pp. 420–425.

[23] G. Georgakis et al. "Multiview RGB-D dataset for object instance detection". In: *2016 Fourth Int. Conf. on 3D Vision (3DV)*. IEEE. 2016, pp. 426–434.

[24] G. Georgakis et al. "Synthesizing training data for object detection in indoor scenes". In: (2017).

[25] I. Goodfellow et al. "Generative adversarial nets". In: *Advances in Neural Information Processing Systems* 27 (2014).

[26] P. Goyal et al. "Accurate, large minibatch sgd: Training imagenet in 1 hour". In: *arXiv preprint arXiv:1706.02677* (2017).

[27] G. Huang et al. "Densely connected convolutional networks". In: *Proc. IEEE conf. on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[28] X. Huang et al. "Multimodal unsupervised image-to-image translation". In: *Proc. European Conf. on Computer Vision (ECCV)*. 2018, pp. 172–189.

[29] P. Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 2017, pp. 1125–1134.

[30] M. Jaderberg et al. "Spatial Transformer Networks". In: *Advances in neural information processing systems* 28 (2015).

[31] D.K. Jain. "An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery". In: *Pattern Recognition Letters* 120 (2019), pp. 112–119.

[32] T. Karras, S. Laine, and T. Aila. "A style-based generator architecture for generative adversarial networks". In: (2018).

[33] D.P. Kingma and M. Welling. *Auto-Encoding Variational Bayes*. 2014.

[34] M.E. Kundegorski et al. "On using feature descriptors as visual words for object detection within x-ray baggage security screening". In: (2016).

[35] J-F. Lalonde and A.A. Efros. "Using color compatibility for assessing image realism". In: *2007 IEEE 11th Int. Conf. on Computer Vision*. IEEE. 2007, pp. 1–8.

[36] D-S. Li et al. "A GAN based method for multiple prohibited items synthesis of X-ray security image". In: *Optoelectronics Letters* 17.2 (2021), pp. 112–117.

[37] C-H. Lin et al. "St-gan: Spatial transformer generative adversarial networks for image compositing". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 2018, pp. 9455–9464.

[38] T-Y. Lin et al. "Focal loss for dense object detection". In: *Proc. IEEE int. conf. on computer vision*. 2017, pp. 2980–2988.

[39] T-Y. Lin et al. "Microsoft coco: Common objects in context". In: *European conf. on computer vision*. Springer. 2014, pp. 740–755.

[40] Y.A Mejjati et al. "Unsupervised attention-guided image to image translation". In: (2018).

[41] D. Mery et al. "Automated X-ray object recognition using an efficient search algorithm in multiple views". In: *Proc. IEEE conf. on computer vision and pattern recognition workshops*. 2013, pp. 368–374.

[42] D. Mery et al. "GDXray: The database of X-ray images for nondestructive testing". In: *Journal of Nondestructive Evaluation* 34.4 (2015), pp. 1–12.

[43] D. Mery et al. "Modern computer vision techniques for x-ray testing in baggage inspection". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.4 (2016), pp. 682–692.

[44] C. Miao et al. "Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images". In: *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 2019, pp. 2119–2128.

[45] S. Michel et al. "Computer-based training increases efficiency in X-ray image interpretation by aviation security screeners". In: *2007 41st Annual IEEE Int. Carnahan Conf. on Security Technology*. 2007, pp. 201–206.

[46] M. Mirza and S. Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[47] E. C. Neiderman and J. L. Fobes. *Threat image projection system*. 2005.

[48] *Oxford University Engineering VGG Annotation Tool*. Online; accessed 12th April 2022. URL: https://www.robots.ox.ac.uk/~vgg/software/via/via-1.0.6.html.

[49] T. Park et al. "Contrastive learning for unpaired image-to-image translation". In: *European Conf. on Computer Vision*. Springer. 2020, pp. 319–345.

[50] D. Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: *Proc. IEEE conf. on computer vision and pattern recognition*. 2016, pp. 2536–2544.

[51] P. Pérez, M. Gangnet, and A. Blake. "Poisson image editing". In: *ACM SIGGRAPH 2003 Papers*. 2003, pp. 313–318.

[52] R. Rizà Porta, Y. Sterchi, and A. Schwaninger. "How Realistic Is Threat Image Projection for X-ray Baggage Screening?" In: *Sensors* 22.6 (2022), p. 2220.

[53] T. Porter and T. Duff. "Compositing digital images". In: *Proc. 11th Annual Conf. on Computer Graphics and Interactive Techniques*. 1984, pp. 253–259.

[54] *PyTorch Open Source Machine Learning Library*. Online; accessed 12th April 2022. URL: https://pytorch.org/.

[55] E. Reinhard et al. "Color transfer between images". In: *IEEE Computer Graphics and Applications* 21.5 (2001), pp. 34–41.

[56] S. Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[57] E. Richardson et al. "Encoding in style: a stylegan encoder for image-to-image translation". In: *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 2021, pp. 2287–2296.

[58] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

[59] S.S.M. Salehi, D. Erdogmus, and A. Gholipour. "Tversky loss function for image segmentation using 3D fully convolutional deep networks". In: *Int. workshop on machine learning in medical imaging*. Springer. 2017, pp. 379–387.

[60] A. Schwaninger, S. Michel, and A. Bolfing. "A Statistical Approach for Image Difficulty Estimation in X-Ray Screening Using Image Measurements". In: *Proc. 4th Symposium on Applied Perception in Graphics and Visualization*. 2007, pp. 123–130.

[61] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In: (2014).

[62] M. Strouhal. "CORSIA-Carbon Offsetting and Reduction Scheme for International Aviation". In: *MAD-Magazine of Aviation Development* 8.1 (2020), pp. 23–28.

[63] K. Sunkavalli et al. "Multi-scale image harmonization". In: *ACM Transactions on Graphics (TOG)* 29.4 (2010), pp. 1–10.

[64] Y. Taigman, A. Polyak, and L. Wolf. "Unsupervised cross-domain image generation". In: (2016).

[65] H. Tang et al. "Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[66] A. Torralba. "Contextual priming for object detection". In: *Int. Journal of Computer Vision* 53.2 (2003), pp. 169–191.

[67] S. Tripathi et al. “Learning to generate synthetic data via compositing”. In: *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 2019, pp. 461–470.

[68] Y-H. Tsai et al. “Deep image harmonization”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 2017, pp. 3789–3797.

[69] T-C. Wang et al. “High-resolution image synthesis and semantic manipulation with conditional gans”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. 2018, pp. 8798–8807.

[70] T.W. Webb et al. “Operationalizing Convolutional Neural Network Architectures for Prohibited Object Detection in X-Ray Imagery”. In: *2021 20th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*. IEEE. 2021, pp. 610–615.

[71] Y. Wei et al. “Occluded prohibited items detection: An x-ray security inspection benchmark and de-occlusion attention module”. In: *Proc. 28th ACM Int. Conf. on Multimedia*. 2020, pp. 138–146.

[72] S. Xue et al. “Understanding and improving the realism of image composites”. In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), pp. 1–10.

[73] Z. You et al. “Imagenet training in minutes”. In: *Proc. 47th Int. Conf. on Parallel Processing*. 2018, pp. 1–10.

[74] F. Zhan, J. Huang, and S. Lu. “Hierarchy composition GAN for high-fidelity image synthesis”. In: *arXiv preprint arXiv:1905.04693* (2019).

[75] F. Zhan, H. Zhu, and S. Lu. “Spatial fusion gan for image synthesis”. In: *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. 2019, pp. 3653–3662.

[76] L. Zhang et al. “Learning object placement by inpainting for compositional data augmentation”. In: *Proc. European Conf. on Computer Vision (ECCV)*. Springer. 2020, pp. 566–581.

[77] X. Zhang et al. “Freeanchor: Learning to match anchors for visual object detection”. In: *Advances in neural information processing systems* 32 (2019).

[78] Y. Zhao, R. Wu, and H. Dong. “Unpaired image-to-image translation using adversarial consistency loss”. In: *European Conf. on Computer Vision*. Springer. 2020, pp. 800–815.

[79] C. Zhu, Y. He, and M. Savvides. “Feature Selective Anchor-Free Module for Single-Shot Object Detection”. In: *Proc. IEEE/CVF conf. on computer vision and pattern recognition*. 2019, pp. 840–849.

[80] J-Y. Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proc. IEEE Int. Conf. on Computer Vision*. 2017, pp. 2223–2232.

[81] J-Y. Zhun et al. “Learning a discriminative model for the perception of realism in composite images”. In: *Proc. IEEE Int. Conf. on Computer Vision*. 2015, pp. 3943–3951.