"JUST" DRIVE : COLOUR BIAS MITIGATION FOR SEMANTIC SEGMENTATION IN THE CONTEXT OF URBAN DRIVING

Jack Stelling *
School of Computing
Newcastle University
Newcastle upon Tyne, UK.
j.stelling2@newcastle.ac.uk

Amir Atapour-Abarghouei †
School of Computing
Newcastle University
Newcastle upon Tyne, UK.
amir.atapour-abarghouei@newcastle.ac.uk

ABSTRACT

Biases can filter into AI technology without our knowledge. Oftentimes, seminal deep learning networks champion increased accuracy above all else. This work attempts to alleviate biases encountered by semantic segmentation models in urban driving scenes, via an iteratively trained *unlearning* algorithm. Convolutional neural networks have been shown to rely on colour and texture rather than geometry. This raises issues when safety critical applications, such as self driving cars, encounter images with covariate shift at test time - induced by lighting, or seasonality. Conceptual proof of bias *unlearning* has been shown on toy, MNIST data; however the strategy has never been applied to the safety critical domain of pixel-wise semantic segmentation of highly variable training data - such as urban scenes. Trained models for both the baseline and bias unlearning scheme were tested for performance on colour-manipulated validation sets showing a disparity of up to 85.50% in mIoU from the original RGB images - confirming segmentation networks strongly depend on the colour information in the training data to make its classification. The bias unlearning scheme shows improvements of handling this covariate shift of up to 61% in the best observed case - and performs consistently better at classifying the "human" and "vehicle" classes compared to the baseline model.

Keywords *Fair AI* · *Bias Unlearning* · *Semantic Segmentation* · *Convolutional Neural Networks*

1 Introduction

Recent years have seen a surge in the development of artificial intelligence (AI) systems; largely fuelled by the symbiosis of deep learning progress [1] [2] [3] [4], and advancements in micro/nanochip manufacturing —harnessing remarkable compute power. Ubiquitous deployment of AI throughout modern society means that practitioners have an ethical and moral responsibility in the dissemination of this cutting-edge technology.

Within the field of deep learning, convolutional neural networks (CNNs) have gained substantial traction in the last decade, and their performance on computer vision tasks is state-of-the-art [5] [6] [7]. Due to the complex nature of these systems, a 'black box' stigma is often attached to them; since deep learning networks are now achieving unprecedented levels of accuracy, external scrutiny and media spotlight demands a push towards transparency, fairness, and accountability [8] [9]. This has led to the more recent movement of 'explainable artificial intelligence' (XAI). To be clear, neural networks can be interrogated for interpretability [10] [11], yielding detailed reverse engineered optimised feature maps, and heatmaps denoting important regions that the model has based its decision on. These solutions tackle the *what* and *where* of the common issues related to explainability, however, we currently can't ask the network *why* it made such a decision.

The crux of developing fair AI is the elimination of bias –a long sought issue in any statistical modelling application. Bias can manifest itself in a multitude of guises which are generally quite context specific. For the purposes of this study we will focus on algorithmic bias. Bias of this nature can creep into our models from training data, meaning our models exhibit the same systemic discrimination found in the wild. This bias can often be unknown and undetected –

*Author †Project Supervisor

making for a particularly insidious force. Indeed, George Santayana warned us that *"those who do not remember the past are condemned to repeat it"*. If bias is allowed to creep into our models undetected, we risk propagating this bias throughout society; eventually distilling into a self-fulfilling prophecy —one which automates inequality.

Natural language processing is particularly susceptible to this prejudice. Training data comprises of swathes of online corpora, and the sheer scale of data means inspecting the training data manually for bias is completely impractical. Sheng et al. [12] demonstrate quite a shocking example of societal stereotyping using a next sentence generation model. Recently, in computer vision fields, it has been shown [13] that software deployed in facial recognition systems perform better on white skinned male individuals rather than other minority races, particularly females. More generally, CNNs have been shown to be biased [14] towards texture (rather than say, shape, colour etc.). Interestingly, the CNNs adopt an Occam's Razor philosophy in their learning process; indeed, if the texture information is sufficient to perform image categorisation well, then why opt for another cue? This becomes problematic when the CNNs pick up on the wrong cues; especially the bias within the data, and use that to make decisions.

(a) Urban Scene Example (b) Semantic Segmentation (c) Instance Segmentation (d) Panoptic Segmentation

Figure 1: Types of image segmentation: **(b)** Pixel-wise categorisation of different semantic classes, e.g. different cars belong to the "car" class and different people to the "people" class. **(c)** Each instance of a *'thing'* has a different label – thus, each person is categorised separately, background, *'stuff'*, classes are ignored. **(d)** A hybrid of (b) and (c). This project will focus on Semantic Segmentation. Exemplar images taken from [15].

This research focusses on colour bias which exists within urban scenes. Urban scene segmentation is at the heart of autonomous vehicle (AV) technology [16], and to date many network architectures have been developed achieving impressive accuracy [2] [4]. This blinkered push towards optimal accuracy often neglects to consider biases within the training data, thus, any advancements in bias mitigation within the realm of image segmentation are novel. Colour bias can manifest in many ways within urban scenes. Successful AV technology must be dependable in a multitude of conditions, including: densely populated urban streets, fog, rain, snow, glare, and seasonal changes. This highly variable distribution of data poses a great challenge. Clearly, the same tree that a segmentation model picks up on in summer might look very different in winter, or in New York would a CNN learn to categorising all yellow boxes as cars due to the number of taxis? We humans are extremely adaptable and can make decisions to correct our actions depending on external stimulus; machines however, struggle to perform well in edge cases or situations not encountered in the training phase. Thus, a push towards robustness and generalisation is paramount for the evolution of safe AV technology.

The umbrella of image segmentation covers three main domains (see Figure 1): instance segmentation [17], semantic segmentation [4] [2], and more recently a hybrid of both – panoptic segmentation [18]. In this work we will consider semantic segmentation, which concerns categorising each pixel in an image into one of n predefined classes. This process splits the image into different regions based on what the pixels show and is the adopted methodology for AV systems.

The aim of this paper is to mitigate colour bias from semantic segmentation models trained on urban driving scenes. The objectives of this project are twofold:

- Firstly, to gain empirical evidence that seminal semantic segmentation architectures do overfit to the colour information in highly variable urban scenes, and, where possible attempt to quantify this.
- Secondly, to determine whether using a multi-headed network architecture, to adversarially remove a *known* bias at train-time in a pixel-wise semantic segmentation model is effective.

2 Related Work

2.1 Segmentation Architectures

For the purposes of semantic segmentation, fully convolutional networks (FCN) have gained the most traction in recent years after the work of Long et al. [19]. FCNs do not have any fully connected, 'dense', layers at the classification head

of the network; solely relying on convolutional layers passed to a classification layer. This preserves spatial information from the input and significantly reduces the number of parameters in the network, thus increasing computational efficiency.

Since semantic segmentation is a pixel-wise classification problem, once the down-sampling process of feature extraction has taken place, the spatial resolution of the feature maps is increased from the bottleneck layer up to the same input size for the classification task. Encoder-decoder style networks achieved state-of-the-art results in 2016 with U-Net [20], SegNet [2] and DeConvNet [21] all adopting the same idea, whereby the first half of the network is a mirror image of the second half, albeit with different methodical nuances. SegNet proposed a computationally fast and memory efficient network via saving the locations (indices) of the maximum values on the max-pooling operation – this is then used during upsampling in the decoder part of the network. U-Net utilised skip connections, a concept used to great effect in ResNet [22], allowing for information from the encoder part of the network to be used in the decoding procedure.

Later, the DeepLab family [23] [24] [4] [5] of architectures pushed the boundaries in semantic segmentation tasks mainly from the application of atrous convolutions [24]; effectively developing the Atrous Spatial Pooling Pyramid (ASPP) module to handle objects of different scales in the same image. ASPP is based on simultaneously performing dilated convolutions with different atrous rates and concatenating the resultant feature maps. This essentially combines multiple fields of view tackling the issue of different scale objects. The DeepLab family have achieved state of the art on benchmark datasets with each iteration of the network.

More recently, the concept of attention has been applied to semantic segmentation tasks [25], creating a more efficient and higher performing model than using multi-scale inference. Similarly, to the problem that DeepLab attempted to solve with the ASPP module, Tao et al. [25] argue that fine detail (bollards, a person in the distance etc.) is often better predicted with a scaled-up image size, whereas large objects (roads, buildings etc.) require more global context and downscaled images are generally more beneficial as the convolutional filters have a larger field of view and thus capture more context. However instead of a concatenation operation merging this information together (as in ASPP) – Tao et al. develop a system whereby prediction for some pixels is performed using the scaled-up images and others use the scaled down images. Further, the ASPP module and other multi-scale context methods e.g., PSPNet [26] are static and not learned; whereas relational methods build context based on image composition. This means that unlike [24] [4] [5] [26], the region of interest using an attention-based mechanism is not restricted to being square – this is advantageous in the context of urban scenes when the geometry is often a product of visual perspective, like a road sign in the foreground covering a long skinny rectangular patch. Tao et al. maintain state of the art performance on the CityScapes [27] and Mallipary Vistas [28] datasets at the time of writing.

In this work we adopt the DeepLabV3 [4] and SegNet [2] architectures to use as baselines for testing novel bias removal concepts in the context of semantic segmentation. DeepLab architectures have been shown [29] to be less susceptible to adversarial attacks, an increasing concern in safety-critical applications such as self-driving cars.

2.2 Bias Removal:

As mentioned in §1, bias, in the context of urban scene segmentation can manifest in many forms. Examples include adverse weather conditions experienced at test time when training data does not account for this, seasonality affecting physical colours (e.g., tree leaves, flora), seasonality affecting lighting/shadows/luminance, more obvious lighting fluctuations from night to day, reflection, shadow, different countries/cultures using different colour systems for highway codes. See supplementary material §A2 for qualitative visual examples of some of these characteristics. This is by no means an exhaustive list, even edge cases such as sporting events, parades and accident blockades can cause out of sample differences that networks must tackle if we are to trust AV technology. These unknown perturbations cause a covariate shift from the input data that the models were trained on, which can cause adverse effects on performance due to the high intercorrelation between network weights.

Large-scale, finely annotated datasets for segmentation are expensive to obtain, requiring human annotation which can often take experienced workers up to 90 minutes an image to complete [27]. Due to this bottleneck, it is not feasible to create site specific training datasets for multiple locations where the cars will operate, thus robust generalisable models must be developed which perform well in a wide variety of situations. Models which can learn bias which exists in the data and account for it are highly desirable. This problem does not only exist within the sphere of AV technology, it also extends to many others, including the fields of augmented reality and virtual reality where indoor scene segmentation is paramount – another task which relies on a highly diverse input distribution.

Under the umbrella of bias removal, the taxonomy forms three natural groupings:

- Those seeking to increase generalisation and thus reduce the effects of bias via image augmentation.
- Those using the network architecture to attempt to remove or mitigate a known bias.

- Those attempting to learn the bias within a given dataset and mitigate it accordingly.

Image augmentation increases variability in the training data by adding perturbations to the input images. This synthetically increases the training set without affecting the information contained. Common perturbations include crops, sheers, flips, and colour jitter. This technique has been well researched in computer vision tasks specifically image classification [30] [31].

Augmentation techniques have been adapted specifically to semantic segmentation where sheering and flipping images may not be the most appropriate approach. Kamann et al. [32] propose the use of a colour mask gained from alpha-blending the ground truth segmentation map with the input data during training, which they coin 'Painting-by-Numbers'. Building on the evidence of Geirhos et. al. [14] who showed that CNNs are biased towards texture, Painting-by-numbers improves the robustness of semantic segmentation models to common image corruptions by making the texture of image classes less reliable and pushing the model to using geometry in the image to perform effective segmentation. This technique does not require more training data, thus is efficient during training. Also motivated by CNN textural reliance, Jackson et al. [33] explore style randomisation via altering colour and texture of the input image using style transfer whilst preserving semantic content, again showing an increase in accuracy.

Multi-head models have been explored [34] [35] [36] posing the ability of networks to *unlearn* a known bias. In fact, it has been shown [36] that some models even exacerbate the biases that are known in the datasets after training is complete, thus, models are using the bias itself as a cue to make a certain categorisation. Wang et. al. demonstrate this with images of women in kitchens -evaluating predictions before and after training on gender-unbalanced datasets. Kim et al. [21] demonstrate a proof-of-concept approach showing that after planting a synthetic colour bias in the MNIST dataset the model uses the obvious colour cue for categorisation rather than geometry of the numbers. A second model head employs an iterative algorithm using reverse gradients to successfully "unlearn" the known colour bias, pushing the model to use the numbers shape for correct classification. This technique has not been applied to highly variable domains such as semantic segmentation, and this work reviews its success.

The contribution of this paper is to increase the robustness of CNNs via the mitigation of algorithmic bias - specifically colour bias found in highly variable urban road scenes. Building on the foundations of Kim et al. [34], we aim to implement an "unlearning" procedure within the network architecture itself rather than increase generalisability via augmentation of the input data. The unlearning procedure employs a multi-headed network to adversarially remove a target bias using reverse gradient loss. Semantic segmentation is used as a vehicle to assess the effectiveness of such a system, albeit the core principle, should, in theory, be able applicable to any deep learning architecture.

3 Removing a Known Bias

This work is most closely aligned with the work of Kim et al. [34] in their paper *Learning Not to Learn* referred to hereafter as LNTL. This proof-of-concept paper synthesised colour bias into the MNIST [37] dataset and successfully used a gradient reversal strategy to remove the colour information from the training data. In real data, we are not afforded the luxury of knowing exactly what the bias is and where it manifests – although it has implicitly been shown that CNNs can pick up on the wrong cues [14]. Thus a comparison between standard training data, a greyscale baseline and an implementation of bias *unlearning* is tested. This strategy is a self-supervised task as we can extract the true colour labels from pixel values of the training data, which we already have.

3.1 Problem Statement

In theory, the set of all images that self driving cars encounter at test time is drawn from one set. This set contains all possible situations that the car could ever encounter, whether it be across a dessert track in midday sun or a snow storm in a mountain pass. Indeed, many landscapes could provide this challenge in a single journey - confirming that this rich theoretical variation is not hyperbole. Practicality dictates that the training dataset is a much restricted subset, and due to this it is not a true representation of situations we *could* encounter in the real world. In this sense, we assume the test set to be unbiased. We aim to train a network on biased data, attempt to systematically *unlearn* that bias during the training phase and then deploy the model to perform on unseen and unbiased test data.

Training and test sets are normally split randomly, in an attempt to eliminate the domain gap between between the two distributions X_{Train} and X_{Test}. Despite best efforts, a gap still can remain, an even larger gap however is the domain gap between X_{Test} and the actual distribution X. For this reason results of the bias mitigation strategy might appear slightly subdued. Analogously, results published for segmentation accuracy are inflated compared to performance in real life. Conveniently the Cityscapes dataset [27] - which consists of dash-cam footage from European cities - is partitioned on city for train, validation, and test sets; which means that model may lean on the nuances apparent in the training cities, making it an ideal candidate to use for this experiment.

Our set-up splits a standard semantic segmentation classification network (e.g. DeepLabV3), into a double-headed network – one with a pixel-wise classification head for the task of semantic segmentation and one with an auxiliary bias classification head. The networks are modular constructs denoted as follows: feature extractor network, $f : \mathcal{X} \mapsto \mathbb{R}^{\mathbb{N}}$, pixel-wise classification network $g : \mathbb{R}^{\mathbb{N}} \mapsto \mathcal{Y}$ and the bias classification head $h : \mathbb{R}^{\mathbb{N}} \mapsto \mathcal{B}$. Where N is the amount of feature maps produced by the embedding network f.

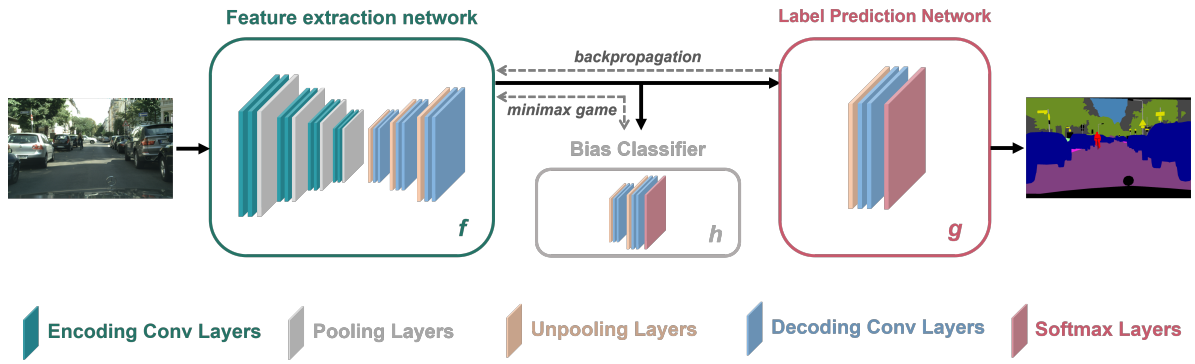


Figure 2: Network architecture showing separate networks *f*, *g* and *h* and their roles in the system.

Figure 2 shows the implemented network architecture, with the sub-networks f, g and h, with the fork depicted in at the last convolutional layer prior to classification. The precise architecture is left intentionally vague because the theory should apply to any system. Specific networks are discussed in more detail in §4.2.

3.2 A Caveat on Fork Placement

Interestingly, since f feeds its outputted feature maps into network g the bias propagates through the network, and so the location of the fork is an arbitrary choice. Furthermore, $f \circ g$ will be void of any bias so long as h has done its job in correctly classifying the bias and the subsequent gradient reversal step successfully discourages f from using such cues. Works focussed on interpretability [10] [11] show that feature maps extracted at the start of a CNN generally contain low level information; often containing blocks of colour and edges, whereas the final layer feature maps contain higher level features with tightly integrated colour information.

Intuitively, we expect that the feature maps towards the end of the network would be the most suitable place to add the fork. The training regime requires a short burst of end-to-end training without including the bias classification network, h. This ensures that the network already has some classifying ability and weights are converging. If we do not allow this head start for the classifying network; when the auxiliary bias network is activated a mode collapse situation could occur, where the network weights are unable to converge towards optimisation [34]. Thus, a fork located towards the end of the main bulk of the segmentation network would allow the weights upstream to be amended whilst the classification layer would be largely unaffected. We hypothesise that a different fork location further upstream would achieve the same result eventually but would take longer to reach convergence. Furthermore, the fork has been added leaving one convolutional layer before the SoftMax layer, this ensures learnable parameters remain in the classification head and allows for any reactive adjustments in network g from a change of it's input, $f(x)$.

3.3 Formulation

The following section guides the reader through the underlying mathematics of the proposed network outlined in §3.1.

Hereafter the following mathematical notation will be adopted. We provide input images, $x \in \mathcal{X}$, with corresponding ground truth labels $y_x \in \mathcal{Y}$ to the initial network f. x in our context are road scene images from the various datasets whereas y_x is a label attributed to each pixel which categorises it into one of n classes. We define a set, $\mathcal{B}$, which contains *all* of the possible biases that $\mathcal{X}$ can contain - the aim of this project is to mitigate colour bias thus $b \in B$ denotes a set of colour predictions where the colour bias $B \subseteq \mathcal{B}$. A hidden mapping exists which maps the the observations from the set $\mathcal{X}$ to the set of biases $\mathcal{B}$, or indeed colour biases, B since this is contained within. We will only consider B in this paper concerning colours - the same concept could be applied to other domains where a different bias was the objective.

We assume X and Y are random variables following the distributions P_X and P_Y respectively whose sample space is $\mathcal{X}$ and $\mathcal{Y}$. x_i, y_i are observations of the random variables X and Y, and $\tilde{x}_i$, $\tilde{y}_i$ are their empirical counterparts.

We wish to minimise a target bias existing within the input training data. This is therefore the same as reducing the mutual information between the feature maps and that bias. Since $\mathcal{I}(X;Y) \to 0$ iff X and Y are independent. That is, we are no more certain about the existence of bias even after observing the predictions and vice versa. So the pixel-wise semantic predictions are independent of colour bias.

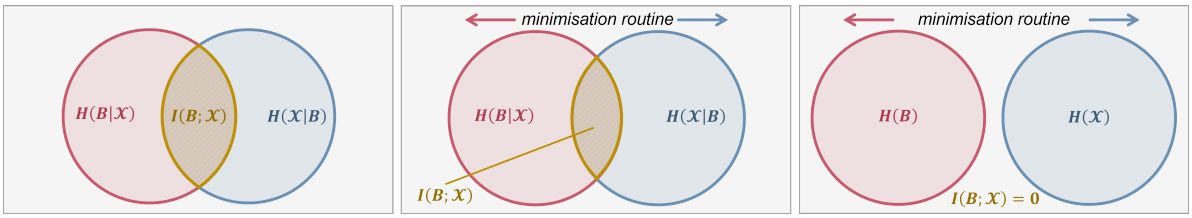


Figure 3: A visual interpretation of a minimisation routine on the mutual information of two random variables: when the mutual information equals zero we have independence.

As discussed in §3.1 we assume that the two distributions of training data and test data aren't congruent and thus bias does exist. Therefore the mutual information between the output of the final classifier and the defined bias can be described as:

$$\mathcal{I}(b(X); g(f(X))) \gg 0, \tag{1}$$

where $g(f(X))$ is the composite function $g \circ f$ representing the normal end-to-end training pipeline of a classifier network. Since g is downstream of f, and accepts the output of f as its input, it is then adequate to attempt to remove the bias from f. This therefore amounts to minimising the mutual information between $f(X)$ and $b(X)$ whilst simultaneously optimising a vanilla classifier:

$$\min_{\theta_f, \theta_g} \mathbb{E}_{\tilde{x} \sim P_{X(x)}} \left[\mathcal{L}_c(y_{\tilde{x}}, g(f(\tilde{x}))) \right] + \alpha \mathcal{I}(b(X); f(X)), \tag{2}$$

where $\mathcal{L}_c(\cdot, \cdot)$ is the cross-entropy loss, and α is a loss-balancing hyperparameter to ensure that one term does not dominate the other. In information theory we can distil the mutual information term in Eq.2 into the relevant marginal and conditional entropies:

$$\mathcal{I}(b(X); f(X)) = H(b(X)) - H(b(X)|f(X)). \tag{3}$$

Intuitively, this is the reduction of uncertainty in $b(X)$ when $f(X)$ is observed. Since the marginal entropy of $b(X)$ is independent of the optimisation parameters θ_f and θ_g we can remove it from the optimisation problem described in Eq.2, leaving only the $-H(b(X)|f(X))$ term. This negative entropy term requires knowledge of the posterior distribution, $P(b(X)|f(X))$, to directly solve – which is intractable in a realistic setting. I.e. we can't sample from all of the possible urban scene images encountered by an autonomous vehicle and calculate the distribution of bias therein. A workaround, known as *Variational Information Maximization* [38], involves obtaining a lower bound via defining an auxiliary distribution $Q(b(X)|f(X))$ to approximate $P(b(X)|f(X))$. In practice we can use the network h to perform this approximation during the training phase. Thus, minimising Eq.3 is reformulated as:

$$\min_{\theta_f} \mathbb{E}_{\tilde{x} \sim P_X(\tilde{x})} \Big[\mathbb{E}_{\tilde{b} \sim Q(\tilde{b}|f(\tilde{x}))} \big[\log Q(\tilde{b}|f(\tilde{x})) \big] \Big]. \tag{4}$$

To quantify the approximate equality constraint that we have had to invoke for Eq.4 to hold, we can introduce the Kullback–Leibler divergence to measure the similarity of two distributions P and Q. The mutual information loss then becomes:

$$\mathcal{L}_{MI} = \mathbb{E}_{\tilde{x} \sim P_X(\tilde{x})} \Big[\mathbb{E}_{\tilde{b} \sim Q(\tilde{b}|f(\tilde{x}))} \big[\log Q(\tilde{b}|f(\tilde{x})) \big] \Big] + \beta D_{KL}(Q(b(X)|f(X)) \parallel P(b(X)|f(X))), \tag{5}$$

where D_{KL} is the KL-Divergence and β is a loss-balancing hyper-parameter. We parametrise distribution Q as our bias predictor network h, denoted θ_h. As h learns in the training phase we would hope that $D_{KL} \to 0$ meaning $Q \to P$. This is the same as empirically training network h with $b(X)$ as the ground truth. To this end we can further relax the latter term of Eq.5 to be the expectation of the cross-entropy of our predictions $h(X)$ from the softmax function of h

and the ground truth colour labels $b(X)$. Note that this formulation would hold for any *known* bias, we would just have to amend the labels $b(X)$ according to the bias we wished to remove. The loss of the bias network is then:

$$\mathcal{L}_{\mathcal{B}}(\theta_f, \theta_h) = \mathbb{E}_{\tilde{x} \sim P_{X(x)}} \big[\mathcal{L}_c(b(\tilde{x}), h(f(\tilde{x}))) \big], \tag{6}$$

If our goal was to solely predict the bias, we would be done. However we want to employ an adversarial strategy where networks f and h play a minimax game. That is, as network h seeks to minimise the loss in Eq.6 we also want to push network f to maximise the loss expressed in the first part of Eq.4; this way the encoder network f attempts to unlearn the colour information, via a gradient reversal strategy and produces corresponding feature maps, whilst the bias classification network uses these updated feature maps and still seeks to push for the best classification. Due to this, during the training scheme we expect the bias loss to begin to converge at the start of training and slowly diverge as training continues, as network f is encoding less and less colour information - rather, it is using different cues for the primary classification task. This loss curve will be a valuable diagnostic.

Holistically, with the original semantic segmentation model added, the problem becomes:

$$\min_{\theta_f, \theta_g} \max_{\theta_h} \mathbb{E}_{\tilde{x} \sim P_{X(x)}} \Big[\mathcal{L}_c(y_{\tilde{x}}, g(f(\tilde{x}))) + \gamma \mathbb{E}_{\tilde{b} \sim Q(\tilde{b}|f(\tilde{x}))} \big[\log Q(\tilde{b}|f(\tilde{x})) \big] \Big] - \beta \mathcal{L}_{\mathcal{B}}(\theta_f, \theta_h), \tag{7}$$

where γ and β are tunable hyperparameters to balance the loss terms. In reality this does not happen simultaneously. The networks are iteratively trained; where we systematically switch networks on and off, leveraging the ability to freeze network weights which are independent of the loss terms in Eq.7, a training schema in §4.3 describes this process in more detail.

4 Experiments

The following section details the experiments undertaken, and provides results and analysis. The structure is as follows: datasets are introduced formally, an explanation of the specific networks used are provided, an explanation of metrics used is given, a detailed training scheme is given whence experiments and results are discussed.

4.1 Datasets

Cityscapes [27] The Cityscapes dataset is a public, and widely used semantic segmentation benchmark. Cityscapes contains data from 50 cities in Germany (and some neighbouring countries) and images are annotated with 30 semantic classes. The dataset contains over 20,000 coarsely annotated images and ≈ 5000 finely annotated images providing pixel-level, instance-level and panoptic semantic ground truth labels. Raw images and segmentation masks are provided in portable network graphics *(.png)* format. Data contains both 16-bit High-Dynamic Range images and 8-bit Low-Dynamic Range format to use at an image size of 1024×2048. We evaluate our semantic segmentation performance on the official 500 annotated validation set.

SYNTHIA [39] The Synthia dataset is also publicly available and comprised of photo-realistic images rendered from a synthetically-rendered virtual city. The dataset used in this paper is the *synthia-rand-cityscapes* subset containing 9400 images at a resolution of 1280×760; which contain labels compatible with Cityscapes, allowing for fairer cross examination of results. The dataset provides fine detail instance segmentation labels - thus data preprocessing was undertaken to create semantic labels from data provided. Synthia images contain very realistic granular detail and complex scenes; some images contain very high numbers of pedestrians - all of which have pixel perfect annotation, leveraging the automatic ability to label images in a generated scene. Further, Synthia has a rich variety of luminance, high scene diversity and vantage differences making it more variable sample space than Cityscapes. Input perturbations during rendering create similar images with slight nuances, due to this some subsequent images appear to have temporal dependence - but they do not. Thus, when creating a 70/30 training/validation split it was decided to split the data as though a temporal dependence did exist within the data. This ensures that similar images do not occur in both the training and validation sets – providing a more representative interpretation of a real setting where a test set is a wholly unseen set of images, this also ensures that we do not get over-optimistic metrics upon evaluation.

4.2 Network Architecture

Since the network f consists of a feature extraction network, we have flexibility to choose any network we like. Indeed, we have positioned this paper for the task of mitigating colour bias in urban scenes, but we could equally employ the technique to other fields of computer vision; say, facial recognition de-biasing.

For the purposes of our primary objective, semantic segmentation, we have chosen to test two seminal architectures, namely, DeeplabV3 and SegNet as discussed in §2. DeeplabV3 is trained with multiple ResNet backbones with ImageNet [40] pretrained weights. As aforementioned this pretraining procedure could add noise to the results as it adds uncertainty about the origins of the CNN bias. Nevertheless, it is an efficiency trade off, and as mentioned in §3.2 anything upstream from the fork can be unlearned. The fork is located directly after the concatenation of the ASPP module within DeepLabV3 and $f(X)$ outputs 1280 feature maps to both the auxiliary bias head h and the primary semantic classifier g.

SegNet follows a simple symmetric encoder-decoder architecture. The encoder is topologically identical to the convolutional layers of the VGG16 [41] network, whilst the decoder upsamples hierarchically by using the indices of the corresponding max pooling operation from the encoding operation. Again the fork for network h is placed before the last convolutional layer and $f(X)$ outputs 64 feature maps to both h and g.

4.3 Training Schema

For a visual representation of Algorithm 1, please see the appendix §A4. Also shown is the method for training the baseline classifier from which our bias 'unlearning' is compared. For a fair comparison of the different methods undertaken, all model hyperparameters remain the same in the segmentation head.

Algorithm 1 Bias removal training schema based on learning not to learn

Require: Training data X, supervised segmentation labels Y, and semi-supervised bias labels H, loaded in batches of size b. Networks f, g, h are parameterised with vectors $\boldsymbol{\theta}$

for each epoch, E_t, **do**
  for each batch, b, **do**
    ▷ # *Stage 1:*
    reset gradients
    produce predictions from segmentation head, $g(Z)$
    produce softmax predictions from bias head $h(Z)$
    calculate manual entropy of bias predictions $\mathcal{L}_B = P(h(Z)) \times P(\log(h(Z)))$
    calculate cross entropy loss in segmentation head, $\mathcal{L}_{Seg}$
    $\mathcal{L}_{Total} = \mathcal{L}_B + \mathcal{L}_{Seg}$
    $\min_{\boldsymbol{\theta}} \mathcal{L}_{Total}$ and update weights
    ▷ # *Stage 2:*
    reset gradients
    produce predictions from segmentation head, $g(Z)$
    produce softmax predictions from bias head $h(Z)$
    while applying gradient reversal upstream from the fork **do**
      calculate cross entropy loss between bias predications, $h(Z)$ and bias labels, H
      minimise loss in bias head
      maximise loss in feature extraction network
      update weights
    end while
  end for
end for

4.4 Metrics

In semantic segmentation dealing with multiclass problems, we often encounter the issue of class imbalance. This occurs when background parts of an image, say buildings or sky, dominate the total pixel count compared to say that of a pedestrian or a red traffic light. Due to this, accuracy becomes an ambiguous metric; inaccuracy of minority classes gets overshadowed by the accuracy of majority classes. Furthermore, in the context of driving scenes the lesser occurring classes are often the most important for human safety. As a result, it is common to use the intersection over union metric (IoU). IoU measures the ratio between the amount of overlap between the predicted and ground truth pixels and the total number of pixels taken up by the prediction and the ground truth, see figure 4 for a visual interpretation.

It is customary to use the mean intersection over union (mIoU) which is quoted in this paper, however, during evaluation we also compute and inspect the individual IoU per category to give a more granular understanding of the networks performance on each class. Scores theoretically fall between 0 and 1, although percentages are often quoted.

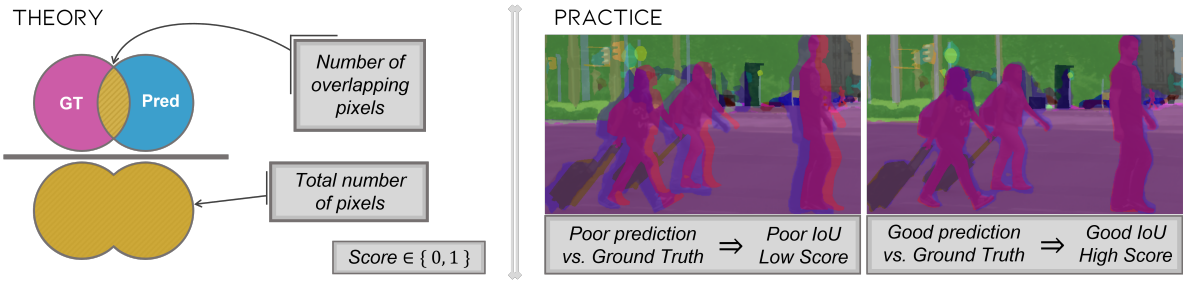
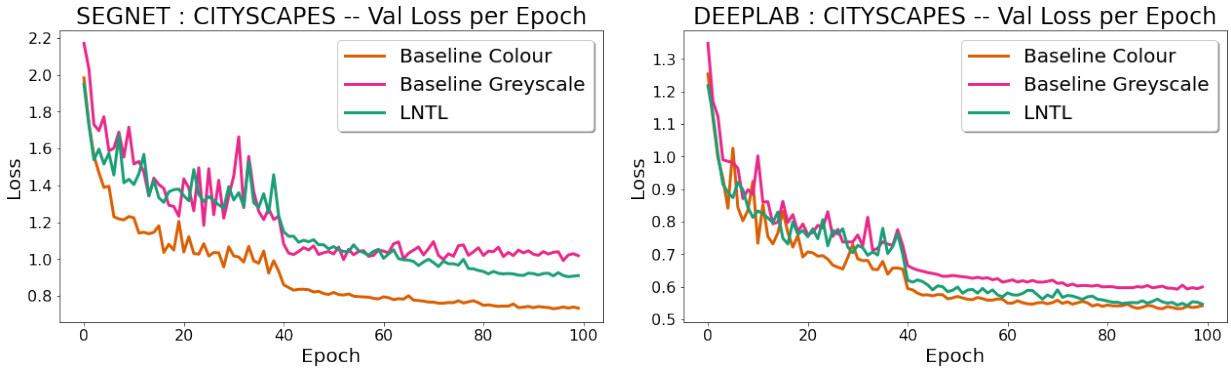


Figure 4: Left pane: Visual demonstration of intersection over union calculation. Right Pane: Predicted segmentation masks from our trained DeepLabV3 model are overlaid on ground truth image masks to demonstrate realistic IOU scores.

4.5 Comparing the Baseline and LNTL Schemes

All models are trained for 100 epochs until convergence. A learning rate of 0.001 was used with the *ADAM* optimiser. Learning rate decay was enforced with the scheduler reducing the learning rate by a factor of 0.1 every 40 epochs. Class weights were computed for all of the training data so the cross-entropy loss function could allow for class imbalance, an extreme occurrence in urban driving scenes.

Baseline models were trained for both SegNet and DeepLab, inputting colour training images and greyscale training images. Since the LNTL scheme penalises the classifier by using the gradient reversal module to adjust the weights, it was expected that the accuracy may suffer somewhat. We did however expect that the LNTL scheme would perform better than the networks making predictions using limited colour information, as is supplied in the greyscale training set.



(a) SegNet Cityscapes Loss

(b) DeepLab Cityscapes Loss

Figure 5: Learning Not To Learn scheme vs. colour and greyscale baseline for SegNet and DeepLab semantic segmentation architectures.

Intuition is confirmed given the loss curves displayed in Figure 5; the LNTL scheme has penalised the loss compared to the converged colour input data for both SegNet and DeepLab architectures. Observe, SegNet minimum loss is 0.730, which DeepLab achieves after just 20 epochs of training. This is a testament to the advancement in the field in only one year between publishing respective networks. Of course, further image augmentation, extra training data and hyperparameter tuning could be used to drive down the loss even further and improve accuracy, however, to satisfy the project hypothesis we only need a comparable canvas to test the concept of applying colour unlearning within the domain of urban scenes. In subsequent experiments the Deeplab network is favoured for its slightly more stable learning, and higher accuracy.

4.6 Synthesising a Covariate Shift in Validation Data

In order to properly test our hypothesis that the LNTL scheme can unlearn colour information in highly variable scenes such as urban images, it is necessary to test both the baseline DeepLab architecture and the LNTL scheme on unseen

data which contains a covariate shift from images it was trained on. To this end three synthesised validation sets are created:

- Converting the validation set to greyscale,
- Adding random colour jitter transformation to the validation images,
- Adding a colour invert transformation to the validation images.

Qualitative examples of these corruptions are available in the report appendix §A3. Models are re-trained on the normal training set, both on SYNTHIA and Cityscapes datasets and only validated on the corrupted images, all other parameters remain the same to allow fair cross-comparison. This enables us to monitor the network loss over the training cycle to see if the LNTL scheme slowly improves in its validation convergence. Monitoring loss in this way allows us to asses overall model health, after all the network is focussed on *minimising* loss not *maximising* accuracy, furthermore we can monitor the loss in the bias head to check for signs of divergence.

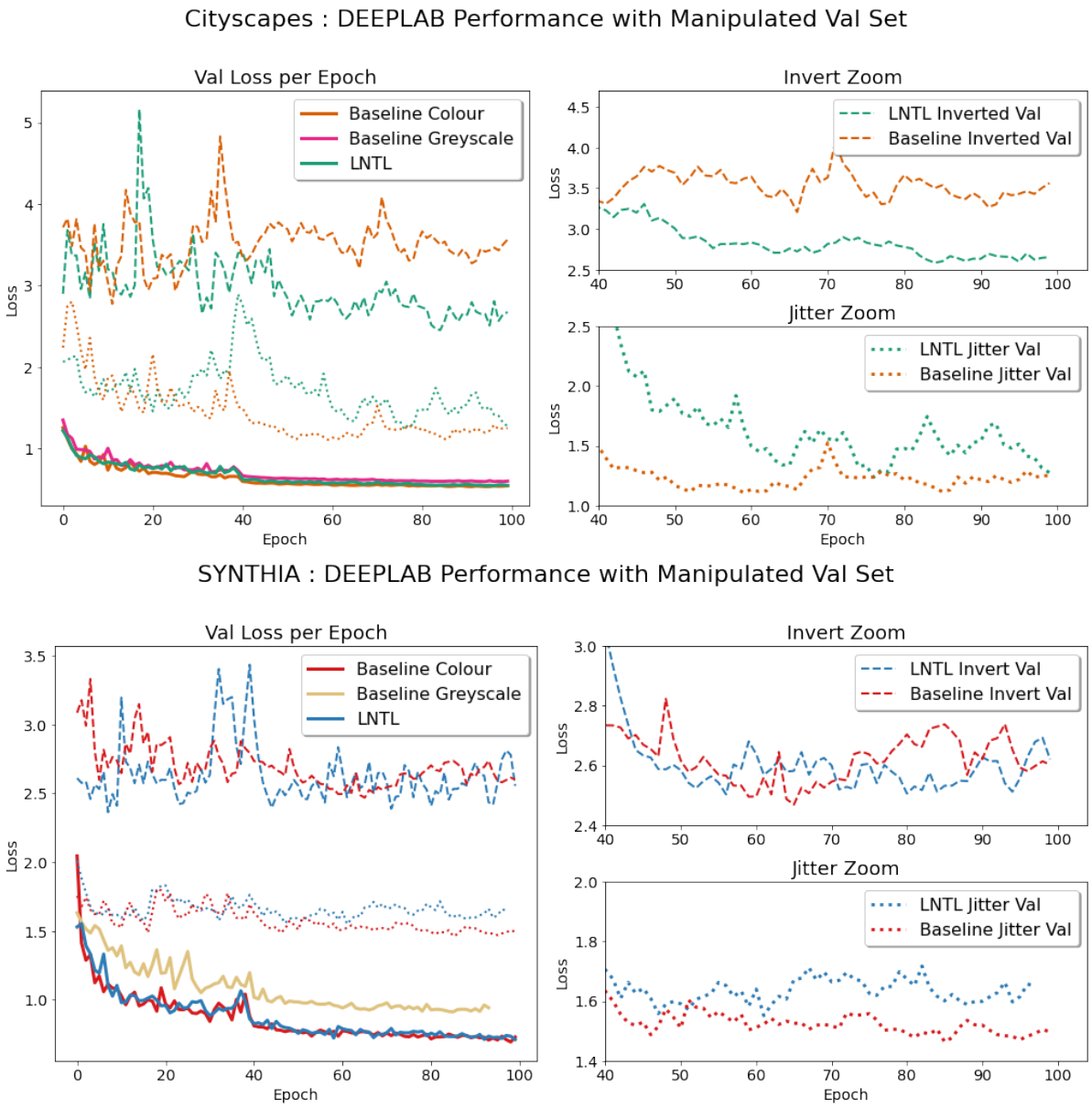


Figure 6: Top Pane: Cityscapes dataset. Bottom Pane: SYNTHIA dataset. Large graph shows the validation results on the normal validation set and the results for manipulated validation sets with colour invert(dashed) and colour jitter (dotted). Adjacent are magnified and truncated plots to the last 40 epochs for the manipulations.

Intuitively, it is worth noting that although greyscale images remove a lot of the colour information, distances between pixel values can still be leveraged, synonymously for colour jitter, despite the stochasticity. Colour invert, however, maximises this difference and corrupts this relationship the most, disrupting spatial inter-correlations between different pixel values.

Figure 6 shows the disparity in loss between uncorrupted validation images and those with perturbed colour; reference lines on the bottom of the graph highlight the extent to which the networks overfit to the colour in the training data. In the Cityscapes dataset the LNTL scheme shows a clear improvement over the baseline method when colour invert is applied to the validation images. This suggests that the LNTL scheme has increased model robustness and has diminished its dependence on colour for categorisation. Although random colour jitter in Cityscapes validation has an increased loss it appears to be converging on a downward trajectory.

Results are not as desirable for the SYNTHIA dataset. Firstly we notice a significant reduction in disparity between uncorrupted and corrupted validation images. This may be due to the rich variety of luminance in SYNTHIA, as mentioned in §4.1. Colour invert seems to affect both schemes equally, whilst colour jitter is marginally favourable to the baseline scheme. As we may expect jitter creates less of a impact on the loss than the invert in all cases.

4.7 Using Synthesised Weather Corruptions as a Proxy for Different Driving Conditions

From the Cityscapes data repository, researchers [42] [43] have created imitations of rain and fog over the normal Cityscapes training data with differing levels of severity. Ground truth labels are exactly the same as for the standard training data so we can leverage this dataset to more closely resemble test images an autonomous vehicle may encounter in the wild. Different severities of image manipulation are provided, we selected one random selection of severity for each image, yielding a 295-image rain validation set and 550-image validation set for fog.

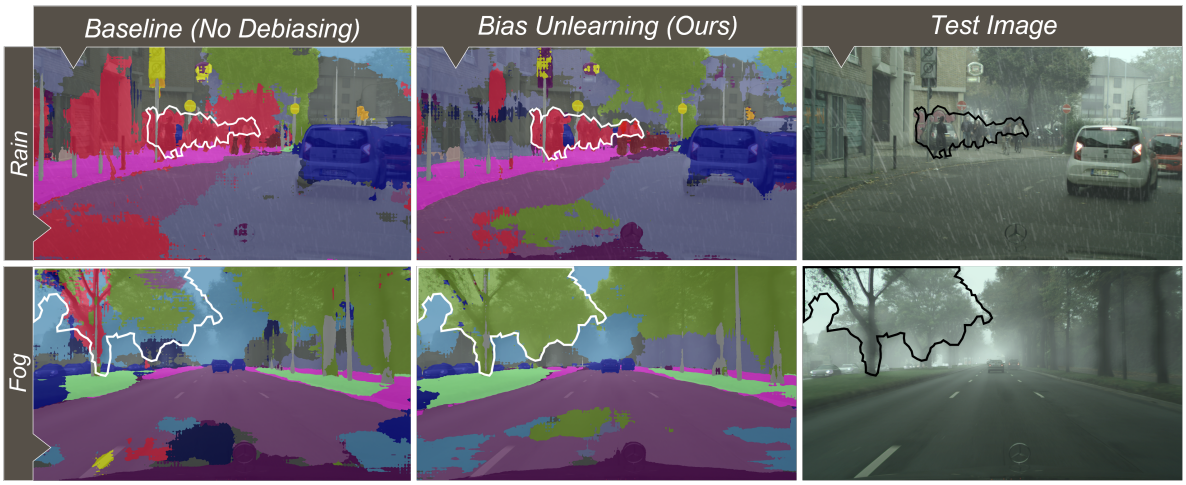


Figure 7: Qualitative results from running the best models displayed in figure 5 for each of baseline, and LNTL. ROIs are shown in bounding boxes for cross-examination. Please see appendix §A5 for more examples of each.

Figure 7 use the synthesized rain and fog validation images fed into the best trained model for each of the baseline and LNTL schemes. All prediction images contain more noise and much poorer performance, as expected from the loss curves in Figure 6. Bounding boxes, showing region of interests, display that the LNTL scheme has managed to correctly segment the pedestrians in the top row scene. The baseline model has produced nonsensical predictions, hallucinating pedestrians on the building in the scene. Similar pedestrian hallucinations can be seen in the tree in the fog scene in row two, and again the LNTL scheme correctly predicting a tree through the fog. The fog image corruptions perform better (*see Table 1*); although noise and artefacts are still present.

4.8 Quantitative Results

Table 1 provides average class mIoU scores for both the DeepLabV3 baseline model and our bias unlearning model. Bold values highlight the best performers for each image transformation. The LNTL scheme not only performs marginally better on the original image, but it performs consistantly better when validated on an out of sample test image - only failing to beat the baseline in the *rain* validation set. This could be due to the relatively small size of the rain validation set compared to others tested.

Table 1: mIoU and mean loss validation results for a DeepLabV3 model trained on Cityscapes training set.

Metric	Scheme	Original	Image Manipulation			Weather Corruption	
		RGB	Greyscale	Invert	Jitter	Rain	Fog
mIoU (%)	Baseline	58.50	36.20	8.50	33.30	**39.40**	52.80
	LNTL *(ours)*	**58.80**	**37.50**	**13.70**	**34.10**	38.90	**54.00**
loss	Baseline	**0.542**	1.173	3.112	1.259	**1.190**	0.873
	LNTL *(ours)*	0.546	**1.156**	**2.423**	**1.180**	1.249	**0.788**

The category-wise and class-wise mIoU scores provide an even more granular understanding of model performance. Both the tables are supplied in the appendix §A5. We highlight some interesting observations from these tables. Firstly the LNTL scheme performs consistantly better in the "human", "nature" and "vehicle" classes than the baseline model. It could be that these categories in particular have a specific, and largely unchanged geometry from scene to scene. I.e. the human form is largely unchanged from individual to individual, furthermore the ASPP module of DeepLab handles objects at different scales. In contrast, the category-wise results show the model performing consistently worse at predicting the "flat" class - perhaps, in the same vein, from buildings having no fixed geometry from scene to scene - largely influenced by viewpoint, suburb etc. At this stage, this interpretation is speculative and further evidence will be sought.

5 Limitations and Future Work

Further work is needed to reinforce the reported findings. A deeper analysis of class-wise mIoU scores would provide more insight into precisely where the bias manifests within images. In particular more granular understanding of the class-wise false-positives and false-negatives of predictions; since in safety-critical applications such as autonomous vehicles this is a vital requisite. I.e. failing to correctly identify a pedestrian has attached with it more gravity than incorrectly classifying a pole, moreover - classifying a pedestrian as "road" has more consequence than classifying a pedestrian as "rider".

Another consideration is analysing the extent to which augmentation techniques e.g [32] interact with the proposed bias unlearning scheme. I.e. does the robustness offered by adequate augmentation reduce the performance observed in this report - or does it compliment it? In addition, urban scene data has high temporal dependence in the wild - a domain of active research and would one which would be interesting to incorporate within our scheme.

On a more horizon view; this project only focussed on the mitigation of a known bias - colour. In the burgeoning world of big data, it is often unsurmountable to assess data for such bias, furthermore we are also at the mercy of our own biased representations. Amini et al. [44] tackle this issue with the use of variational autoencoders (VAEs). The proposed model actually learns, in an unsupervised manner, the latent structure of the input data and adaptively uses this learned latent distribution to selectively upsample underrepresented data points. This allows the model itself to determine the bias inherent within the data at train time. Although Amini et al. demonstrate this technique through racial and gender bias in facial recognition systems the idea itself is generalisable to multiple domains. We have shown that colour bias does exist, the inter-correlation between variables in the input distribution may be more sophisticated than simply penalising a colour value - thus this scheme would get the model itself to do the leg-work.

6 Conclusion

We applied a colour bias unlearning scheme to highly variable images of urban road scenes as an iterative learning process at train-time. Our contribution empirically shows that semantic segmentation architectures do overfit to the colour within training data, and they struggle to generalise to unseen test data –even from a very similar input distribution, as seen in raw $\rightarrow$ weather manipulation experiments. In the worst case; when validating on a set with a colour invert transformation, reductions of 85.50% were observed. We demonstrate that the *unlearning* technique itself is viable, showing a qualitative improvement to both *stuff* and *things* classes in pixel-wise semantic segmentation, from a benchmark seminal architecture - mIoU metrics confirmed this improvement. We observed a 62% increase in mIoU score for colour invert; when neglecting the result for colour invert, we still observe an average increase of 1.5% over all validation set manipulations tested. Furthermore, an average increase of 14.5% was observed for the "human" class, enhancing pragmatic performance in a safety-critical application such as autonomous driving. We position this paper to push towards robust, trustworthy technology - aiming for a transparent and explainable future in artificial intelligence, alleviating algorithmic bias.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. Vol:39, 2481–2495, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Computer Vision – ECCV*, pp. 833–851, 2018.

[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114 , 2019.

[7] R. P. C. Kumar B. and Mohana, "Yolov3 and yolov4: Multiple object detection for surveillance applications," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1316–1321, 2020.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[9] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3449–3457, 2017.

[10] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," in *Workshop at International Conference on Learning Representations*, 2014.

[11] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding Neural Networks Through Deep Visualization," in *the Deep Learning Workshop, 31st International Conference on Machine Learning*, 2015.

[12] E. Sheng, K.-W. Chang, P. Natarajan, and N. Peng, "The Woman Worked as a Babysitter: On Biases in Language Generation," in *the 9th International Joint Conference on Natural Language Processing, IJCNLP*, pp. 3398–3403, 2019.

[13] J. Buolamwini and T. Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," in *Proceedings of Machine Learning Research, 2018 Conference on Fairness, Accountability, and Transparency*, pp. 81:1–15, 2018.

[14] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *ICLR*, 2019.

[15] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9404–9413, 2019.

[16] J. Hawke, V. Badrinarayanan, A. Kendall *et al.*, "Reimagining an autonomous vehicle," *arXiv preprint arXiv:2108.05805*, 2021.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[18] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241, 2015.

[21] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[23] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[24] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[25] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical Multi-Scale Attention for Semantic Segmentation," in *arXiv preprint arXiv:2005.10821*, 2020.

[26] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

[28] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[29] A. Arnab, O. Miksik, and P. H. S. Torr, "On the Robustness of Semantic Segmentation Models to Adversarial Attacks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 888–897, 2018.

[30] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty," in *ICLR*, 2020.

[31] J. Zhang, Y. Zhang, and X. Xu, "ObjectAug: Object-level Data Augmentation for Semantic Image Segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[32] C. Kamann, B. Güssefeld, R. Hutmacher, J. H. Metzen, and C. Rother, "Increasing the Robustness of Semantic Segmentation Models with Painting-by-Numbers," in *Computer Vision – 2020 ECCV*, pp. 369–387, 2020.

[33] P. T. Jackson, A. Atapour-Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style Augmentation: Data Augmentation via Style Randomization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[34] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, "Learning Not to Learn: Training Deep Neural Networks With Biased Data," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9004–9012, 2019.

[35] M. Alvi, A. Zisserman, and C. Nellaaker, "Turning a Blind Eye: Explicit Removal of Biases and Variation from Deep Neural Network Embeddings," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[36] T. Wang, J. Zhao, M. Yatskar, K.-W. Chang, and V. Ordonez, "Balanced Datasets Are Not Enough: Estimating and Mitigating Gender Bias in Deep Image Representations," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5309–5318, 2019.

[37] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[38] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

[39] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR 2015*, 2015.

[42] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," in *ECCV 2018*, pp. 973–992, 2018.

[44] A. Amini, A. P. Soleimany, W. Schwarting, S. N. Bhatia, and D. Rus, "Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 289–295, 2019.