Fashion Images Background Removal with Transformers

Riccardo Mancini, Enrico Tarsi

***Abstract*—This article presents two methods for identifying and removing background from fashion images using advanced semantic segmentation techniques. Two different neural network based on transformers have been used. In contrast to convolution-based methods, these approach allows to model global context already at the first layer and throughout the network. The reference dataset for this project consisting of 45,600 fashion images, each containing at least one cloth, not necessary worn by person, and a .csv file containing the encoded pixel, for every image, to generate the masks. After that, every single image with his mask generated was given in input to some chosen architectures of Segmenter and SETR transformer-based network. These networks was pre-trained on ADE20K dataset to avoid the training on the fashion dataset from scratch. To evaluate the models after training, the metrics used were IoU (Intersection over Union) and Pixel Accuracy, the most common evaluation metrics for semantic image segmentation. In the end, if the results obtained were satisfactory, it procedeed with the removal of the background from another set of images other than those used for training phases.**

I. Introduction

Background removal is the most frequently used photo manipulation technique in post processing tasks. This kind of manipulation is very important for several areas, in particular for e-commerce. In fact it allows the consumers to have a qualitative and accurate visual experience and furthermore they will be focused on the main products without distractions or any type of eye stress. This task is quite easy to do manually, but an automatic approach that allows to do it on a large scale and in short time is a challenging task.

Deep Learning, with all Computer Vision tasks, are the key technology behind automatic approaches. Examining the various tasks they offer that might work for this project, it is easy to see that the best option is the Semantic segmentation task. The concept behind semantic image segmentation is that each pixel of an image belongs to a particular class and receives that class's label; this allows to cluster parts of an image together which belong to the same object class. The abstract idea about the method is that when each pixel is given a class label, the neural network is trained over numerous examples to learn what kind of pixel belongs to a particular class and which pixel belongs to the other class. After the pixel classifier is trained, the resulting pixels that belong to background class are removed.

The background removal technique described in this paper is oriented to the fashion sector. To obtain acceptable results, have been studied all related works presented in Section II, with their prons and cons, in order to take a step forward in this sector. Section III discusses the methods used for the reference dataset containing fashion images, furthermore explore the networks chosen for the purpose discussed and the training settings used for prepare architectures. Finally are shown some metrics used for testing them. Section IV presents all the results obtained and the related discussions. The conclusions, with possible future improvements, are presented in Section V.

II. Related Works

There are several methods to perform semantic segmentation. One of the most famous classical computer vision technique for background removal is segmentation-based color thresholding[1], the whole image is divided into multiple smaller segments, and each pixel value is compared with a previously set threshold. The threshold suppresses the background information by reducing the intensity value of the pixels to that of a single solid shade of grey. Most recent approaches are based on deep learning techniques. The first models building of semantic segmentation are UNet[2] and DeepLab[3].

In 2015 was released UNet that builds on top of the fully convolutional network from above. It was built for medical purposes to find tumours in the lungs or brain. It consists of an encoder which down-samples the input image to a feature map and the decoder which up-samples the feature map to input image size using transposed convolutional layers. The main contribution of this architecture is the shortcut connections that propose to solve the information loss problem due to the processed images. It proposes to send information to every decoder up sampling layer from the corresponding encoder down sampling layer, thus capturing finer information whilst also keeping the computation low. Since the layers at the beginning of the encoder would have more information they would bolster the up sampling operation of decoder by providing fine details corresponding to the input images thus improving the results a lot. Since 2018, UNet has gained huge popularity and been used in some form or the other for several different tasks related to segmentation. In fact, in 2020 was build one variant of U-net called U²-Net[4] or U-squared Net. U²-Net is basically a U-Net made of U-Net. The design has the following advantages: it is able to capture more contextual information from different scales thanks to the mixture of receptive fields of different sizes (ReSidual U-blocks (RSU)); it increases the depth of the whole architecture without significantly increasing the computational cost.

Looking at others networks, in 2017 was published as a deep learning model for semantic image segmentation. Spatial pyramid pooling module or encoder-decoder structure are implemented in various architectures of this model. The former networks are able to encode multi-scale contextual information

by probing the incoming features with filters or pooling operations at multiple rates and multiple effective fields-of-view, while the latter networks can capture sharper object boundaries by gradually recovering the spatial information. Combining the advantages from both methods give birth to the latest architecture of this network: DeepLabv3+[5]; it extends previous version (DeepLabv3[6]) by adding a simple yet effective decoder module to refine the segmentation results especially along object boundaries.

As it can see, most recent semantic segmentation methods adopt a fully-convolutional network with an encoder-decoder architectures. The encoder progressively reduces the spatial resolution and learns more abstract/semantic visual concepts with larger receptive fields. Since context modeling is critical for segmentation, the latest efforts have been focused on increasing the receptive field, through dilated/atrous convolutions or inserting attention modules. Such methods, however, still rely on convolutional backbones and are, hence, biased towards local interactions due the local nature of convolutional filters, limiting the access to the global information in the image. To overcome these limitations, it was formulated the problem of semantic segmentation as a sequence-to-sequence problem and using transformer architecture to leverage contextual information at every stage of the model. By design, transformers can capture global interactions between elements of an image. In 2021, these concepts were implemented with publication of two types of model transformer based, called: Segmenter[7] and SETR[8].

III. Materials and Methods

This section analyzes the methodologies used for the study of the problem. Section A describes the dataset and the annotations. Section B presents the architectures used for the semantic segmentation task. Training settings are exposed in Section C. At last, Section D lists the metrics used for evaluation.

A. Dataset

The dataset is the iMaterialist (Fashion) 2020 used for the competition: iMaterialist Challenge (Fashion) at FCVC7 2020. This dataset contains images of people wearing a variety of clothing types in a variety of poses, in daily-life, celebrity events, and online shopping. It consists of approximately 40,000 images and corresponding fashion/apparel segmentation. This dataset contains 46 apparel objects (27 main apparel items and 19 apparel parts) and 294 related fine-grained attributes, for this task only the main apparel items were considered as classes and the fine-grained attributes were not used. Images are named with a unique ImageId while the annotation are available in .csv format; each row represents a single predicted apparel object/part segmentation for the given ImageId, and predicted ClassId, and each mask is provided using run-length encoding on pixel values. It was necessary to merge the masks that have the same ImageId so as to achieve a single mask for each ImageId.

The dataset was divided into 70% Training Set, 20% Validation Set and 10% Test Set.

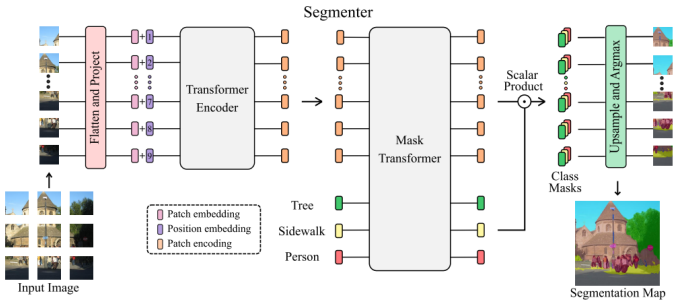


Fig. 1: Segmenter structure

B. Segmentation using transformers

Segmentation semantic task has been implemented with two transformers deep learning model using different architectures. Instead of building these models from scratch to solve a similar problem, have been used the models trained on other problem as a starting point. In fact, the architectures taken, was pretrained on ADE20K dataset.

1) Segmenter[7]: Segmenter is based on a fully transformer-based encoder-decoder architecture mapping a sequence of patch embeddings to pixel-level class annotations. Figure 1 show the network's structure. As it can see, the sequence of patches is encoded by a transformer encoder and decoded by either a point-wise linear mapping or a mask transformer. The encoder was build upon the Vision Transformer model (ViT)[9] used for image classification that employs a Transformer-like architecture over patches of the image previously obtained; among all the existing variants, the following ViTs were considered: "Tiny" and "Base" models. For the decoder was used a Mask Transformer, where the sequence of patch encodings is decoded to a segmentation map. The decoder learns to map patch-level encodings coming from the encoder to patch-level class scores. Next these patch-level class scores are upsampled by bilinear interpolation to pixel-level scores using a softmax followed by a norm; these scores form the final segmentation map.

Segmenter model is trained end-to-end with a per-pixel cross-entropy loss. At inference time, argmax is applied after upsampling to obtain a single class per pixel.

2) SETR[8]: SETR is based on a pure self-attention encoder combined with a simple decoder to provide a powerful segmentation model. First of all,the input image is splitted into fixed-size patches, linearly embedded each of them, added position embeddings, and feeded the resulting sequence of vectors to a standard Transformer encoder. Concretely, the Transformer, as depicted in Figure 2, accepts a 1D sequence of feature embeddings as input. There are two variants of the encoder "T-Base" and "T-Large" with 12 and 24 layers respectively. In the architectures used for this project was taken the second one. To perform pixel-wise segmentation, are available three different decoder designs:

- Naive upsampling that adopt a simple 2-layer network with architecture: 1 × 1 conv + sync batch norm (w/ ReLU) + 1 × 1 conv. After that it upsample the output to the full image resolution, followed by a classification layer with pixel-wise cross-entropy loss.

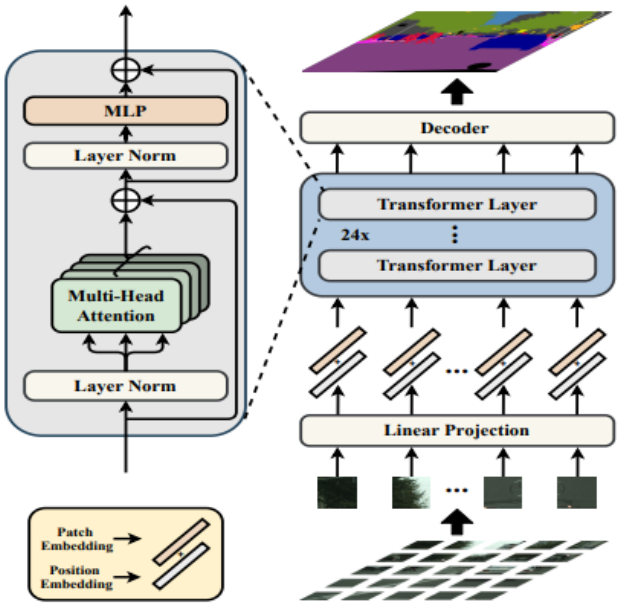
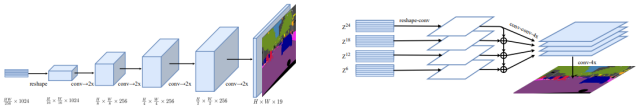


Fig. 2: SETR structure



(a) SETR-pup decoder

(b) SETR-mla decoder

Fig. 3: Two variants of SETR's decoder

- Progressive upsampling (resulting in a variant called SETR-PUP) that introduce a progressive upsampling strategy that alternates conv layers and upsampling operations (Figure 3a).
- Multi-level feature aggregation (a variant called SETR-MLA) that is characterized by multi-level feature aggregation when the feature representation of every SETR's layer share the same resolution without a pyramid shape (Figure 3b).

The last two decoders represented in figure 3 were integrated in the architectures used for the project.

C. Training Settings

During training, it was followed the standard pipeline from the semantic segmentation library MMSegmentation which provides mean substraction, random resizing of the image to a ratio between 0.5 and 2.0 and random left-right flipping. Then, has been applied a randomly crop large images and pad small images to a fixed size of 512×512, that was the dimension used by models pre-trained on ADE20K dataset.

Each architecture has done a distributed training of 15 epochs across 2 GPUs RTX 2080. The batch size for Segmenter ViT-T and ViT-B has been set respectively of 8 and 4. Instead, the batch size of SETR-PUP and MLA has been set for both to 2. Transformer based models have an higher memory occupation and it was not possible to choose a larger batch size. A Vision Transformer encoder accepts in input a fixed-size patches of input images, so the size of these patches has been set to 16 to have low impact on GPU's memory in this case as well.

To fine-tune the pre-trained models for the semantic segmentation task, it was used the standard pixel-wise cross-entropy loss without weight rebalancing. Futhermore, it was employed the stochastic gradient descent (SGD) as the optimize, with a "polynomial" learning rate decay, starting with a learning rate set to 0.001 and a momentum set to 0.9 for all architectures.

D. Performance Metrics

The metrics used to evaluate the performance of the networks were the Average Pixel Accuracy and mIoU.

- Average Pixel Accuracy: This is the simplest method to evaluate how well an image segmentation model performs. In order to do so, calculated, compared to each class in the image, the True Positive (pixel classified correctly), True Negative (pixel classified incorrectly), False Positive (pixel classified correctly), and False Negative (pixel classified incorrectly) values.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

  The average Pixel Accuracy is calculated by taking the pixel accuracy of each class and averaging them.
- mIoU: The IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth.

$$IoU = \frac{AreaOfOverlap}{AreaOfUnion} \tag{2}$$

  The mean Intersection over Union is calculated by taking the IoU of each class and averaging them.

IV. Results and Discussion

Model	Encoder	Decoder	aAcc	mIoU
Segmenter	ViT-T	Mask Transformer	92.35	38.64
	ViT-B	Mask Transformer	94.76	47.11
SETR	ViT-L	MLA	94.42	42.59
	ViT-L	PUP	94.22	41.23

TABLE I: Table of results

Figures 4 and 5 compares the trend of loss and accuracy calculated during the training phase. The models Segmenter ViT-B, SETR-MLA and SETR-PUP have achieved excellent results, in particular Segmenter ViT-B and SETR-MLA, instead the Segmenter ViT-T model show high loss value despite good accuracy as it this model is too simple to extract required features for prediction.

It was necessary for resource reason to execute the validation phase post-training. For all architectures it was calculated the mIoU on the validation-set for each epochs. Looking the figure 6 in both SETR models occurs overfitting and

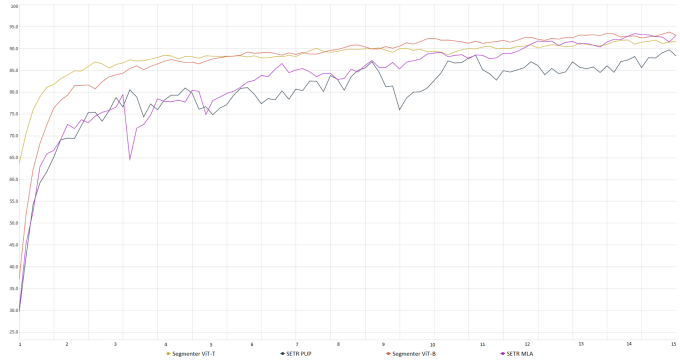


Fig. 4: Train accuracy

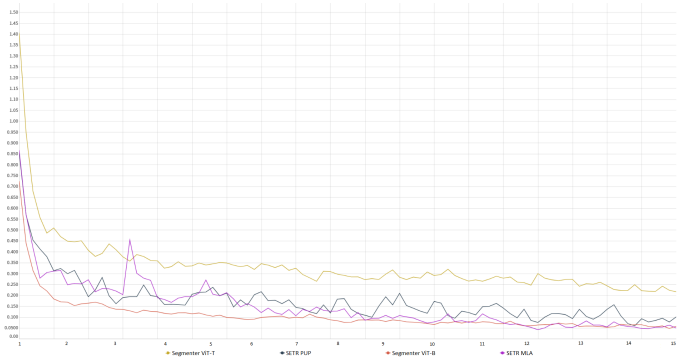


Fig. 5: Train loss

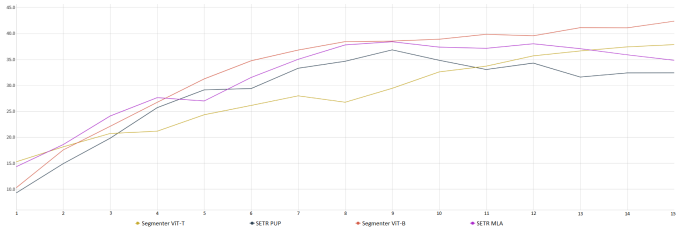
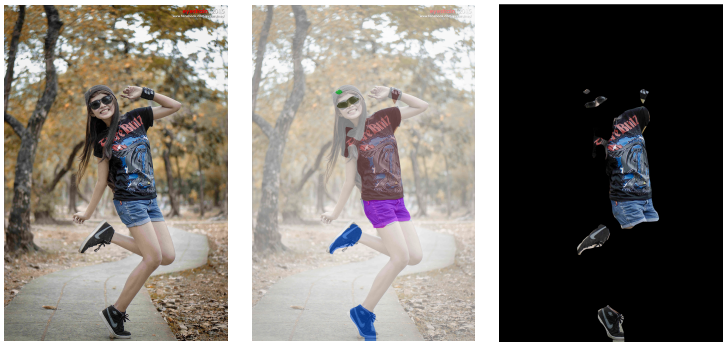


Fig. 6: Validation mIoU



(a) Original image (b) Segmentation map (c) Background removed

Fig. 7: Inference test

we can see an irregular trend, so for SETR-MLA we chose weights from epoch 8 and for SETR-PUP from epoch 9. These behaviors are attributable to too small batch size of 2, indeed these doesn't occurs on Segmenter models.

Whereas these kind of architectures based on ViT are very heavy in terms of dimension and they require a lot of computational power for the training phase, the results achieved during the testing phase reported in Table I can be considered pretty good for all the networks. We can see that SETR models trained with batch size of 2 get comparable results to Segmenter models trained with a batch size of 4 and 8. This shows, as we expected, that the encoder based on ViT-L used in SETR models is more performing than the encoder based on ViT-T and ViT-B used in Segmenter models. Both metrics used mIoU and aAcc confirms that the best architecture is Segmenter ViT-B.

As it can see from the inference done by Segmenter ViT-B in figures 7, that has the best mIoU compared to the others, unlike the main classes like shorts and t-shirt, the classes of small object such as clock and hat are difficult to identify for the network. This behavior has a big impact when we calculate the mean IoU that can involve a significant reduction while the aAcc is not effected.

V. Conclusion

Looking at the output examples produced, the approach offered in this paper has proved to have a very good potential as a background removal in fashion sector. To improve evaluation performance, more hardware resources are needed to increase at least the batch size parameter used on training phases. Due to the nature of transformer-networks based, they require to see many samples at time during every epoch of training, to learn how to generalize as best as possible in sets of data never seen. Another improvement that could lead to higher quality of background removal is the decrease of patch size to 8. This approach would increase the ability of ViT to extract even better feature from input images. Even though it was reached similar performances by Segmenter ViT-B and SETR-MLA architectures, future works could be based on the modification of the structure of Segmenter's encoder, replaced by the larger one: ViT-L (Large), as used in the SETR's structure, for a fair comparison. One last observation to be able to implement the general performances could be related to the unbalance training data, where the number of examples in the training dataset for each class label is not balanced. That is, where the class distribution is not equal or close to equal, and is instead biased or skewed. There are different techniques to overcome this problem like oversampling data, that consist of creating artificial data points or duplicates of the minority class sample to balance the class label (data augmentation); or another method could consists to tune the function loss during training phase in order to obtain that the class weight becomes inversely proportional to the class frequency in the input data.

References

[1] N. Kulkarni, "Color thresholding method for image segmentation of natural images," *International Journal of Image, Graphics and Signal Processing*, vol. 4, no. 1, p. 28, 2012.

[2] 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*. Springer, 2015, pp. 234–241.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[4] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, p. 107404, 2020.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[7] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7262–7272.

[8] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[9] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE transactions on pattern analysis and machine intelligence*, 2022.