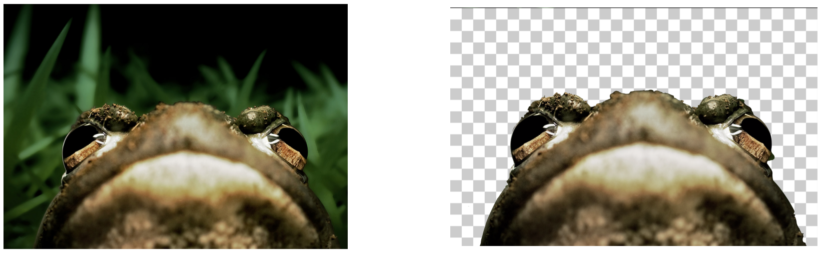
APS360 - *ExtractNet* Project Proposal

Mohsin Hasan, Nahian Khan, Sagar Patel

March 11, 2019

1 Introduction

We aim to create *ExtractNet*, a software tool which removes backgrounds from images that have central and prominent objects. This is an application of image segmentation, which is the clustering of pixels into salient regions [1]. The segmentation of images into separate regions allows for the extraction of prominent objects.



(a) With background.

(b) Without background.

Figure 1: Example of background removal. Images modified from [2].

There are several motives for background removal. For example, the media industry can use background removal to refine publication images, and the autonomous vehicle industry can use it to enhance machine vision. The rapid growth [3][4] of both of these industries highlights the importance of the task.

Normally, one would have to use tools such as Adobe Photoshop to extract backgrounds manually [5]. However, this process can be time consuming, requires expertise and cannot be done in real-time. Machine learning is therefore an excellent tool for automating the task. Given enough data, machine learning approaches have been shown to outperform traditional algorithms on similar segmentation tasks [6].

2 Source of Data

Data will primarily be gathered from the Common Objects in Context (COCO) dataset since it provides masks highlighting object classes in images [7][8], as shown below. These serve as a “ground truth” for segmentation.

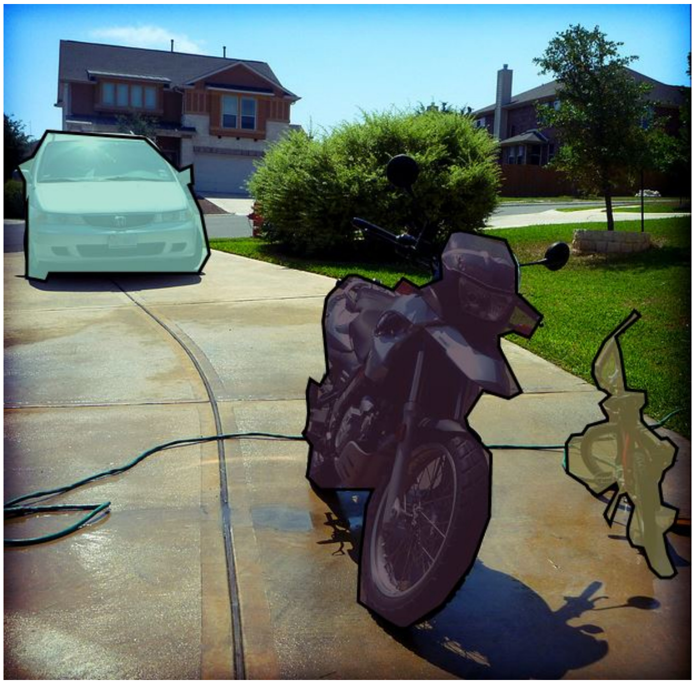


Figure 2: Example of COCO image with object masks [8].

The data cleaning involves the following steps:

- From the dataset, extract images which contain up to 4 primary objects from the super categories: 'vehicle', 'animal', or 'person' [7]. Together, the objects should compose over 30% of the image. This is because these categories often appear together in images, and (if large enough) are clearly distinguishable from the background to humans.
- Combine the masks into a single, 1 channel mask, highlighting only the chosen primary objects. This is because our task does not need the masks of other extraneous objects, and the mask format is used for training.

3 Overall Structure of Software

Below is a block diagram of *ExtractNet*, and how it is intended to function on an example image.

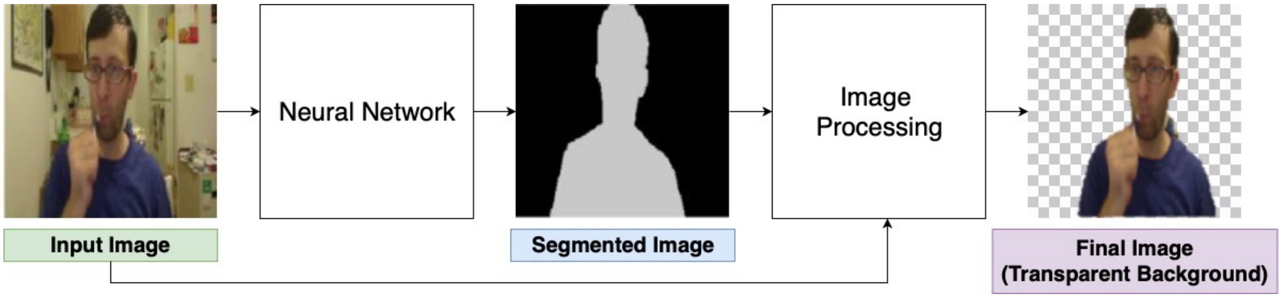


Figure 3: A block diagram of our software. Images modified from [9].

3.1 Neural Network Block

This block classifies the pixels composing the primary objects in a given image. It takes as input the original image, and produces a single channel, mask image of the same size as output.

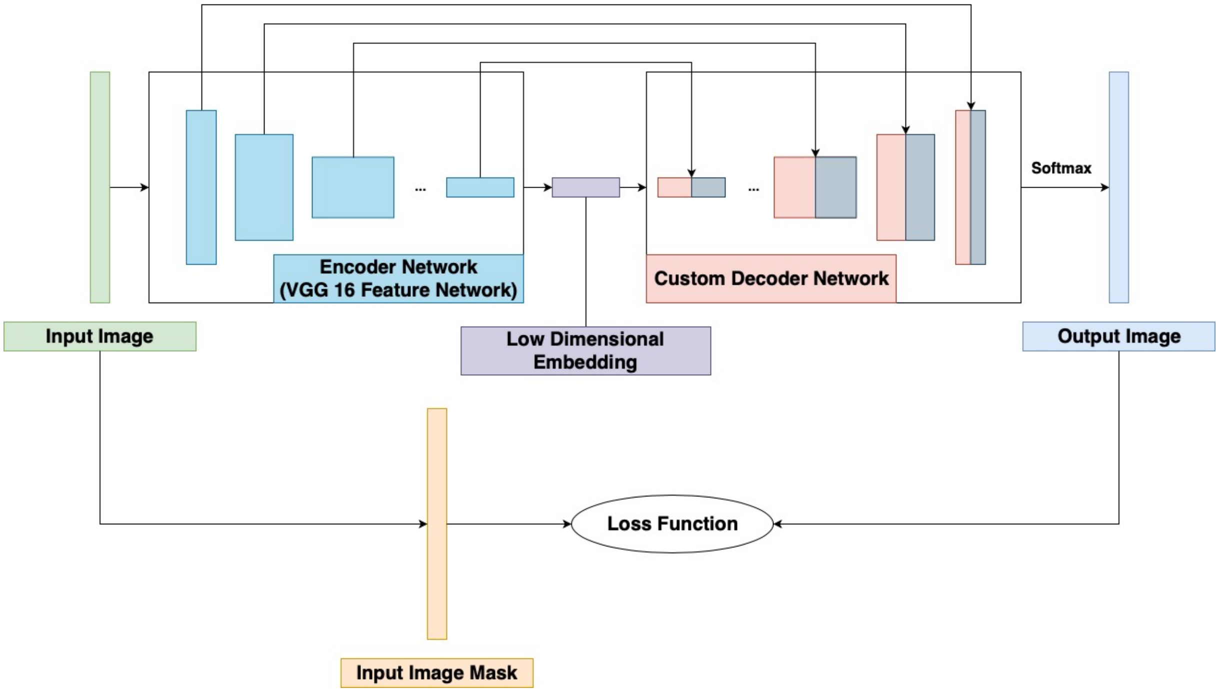


Figure 4: The neural network in training. The blocks depict a combination of convolution, pooling, and ReLU activations.

The intended architecture involves a mirrored, fully convolutional, encoder-decoder pair. The fully convolutional architecture allows arbitrary input sizes (so long as dimensions are divisible for downsampling). The architecture is based on TernausNet, which uses a pre-trained encoder for image segmentation [10].

- Encoder: The encoder network is the VGG-16 feature network (convolutional parts), which will be used with their pre-trained weights [11].
- Decoder: The decoder network is custom built, and reverses the encoder's steps to transform the embedding into a full image of the same size as the input. Importantly, a layer in the decoder uses not only the previous layer's output, but also concatenates the output of the corresponding encoder layer, intuitively, to create a more accurate mask. The image is passed through a softmax layer to generate the probabilities for each pixel being part of a primary object. The decoder network is trained by computing the binary cross entropy loss between this and the "ground truth mask" (since this task involves pixel-wise binary classification).

3.2 Image Processing Block

This block uses the output of the previous block to remove the background.

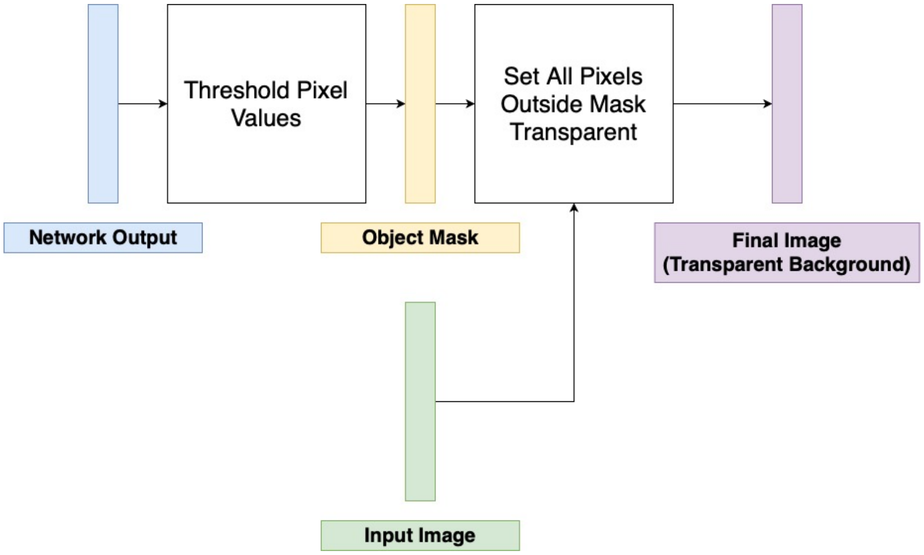

Threshold Pixel Values
Set All Pixels Outside Mask Transparent
Network Output
Object Mask
Final Image (Transparent Background)
Input Image

Figure 5: Internal diagram for the image processing block

In order to produce a solid object mask, the probabilistic pixel values are rounded to binary values (where 1 indicates the pixel belongs to a primary object). This mask is then applied to the input image to isolate the primary object, and make the background transparent, producing the final image.

4 Plan

Task	**Est. Time** (hr/person)	**Deadline** (mm/dd)
Data Cleaning & Partition Scripts	5	03-02
Training Environment Setup (Google Colab.)	1	03-07
Initial Model Training (Overfit to smaller dataset to test functionality)	4	03-07
Baseline Model Implementation	2	03-10
Hyperparameter Tuning	7	03-14
Progress Report	4	03-17
Software Integration (Image Processing Block)	2	03-21
Presentation Slides Draft & Practice	4	03-28
Presentation Slides Finalization & Submission	5	03-29
Project Report	6	04-05

5 Risks

Some of the risks associated with this project are:

- The VGG-16 feature network was trained on a specific image size (224x244) and since we don't adjust its weights, our software may not perform well on input images of arbitrary size. Larger image sizes may also result in long computation times.

  Possible Solution: We can crop and resize input images (and masks) to 224x224, extract the primary objects, and revert the result to its original size. This potentially causes a more blurry result, but may improve performance overall (and computation times).

- The decoder network is as deep as VGG-16 (Encoder) and can lead to long training times and difficulty in debugging.

  Possible Solution: Smaller datasets can be used during initial stages before training on the full dataset. If the training time is still intractable, simpler architectures may be considered.

- Since the data cleaning stage involves multiple per-image steps, with over 100K images to parse through, it may be computationally expensive, and exceed the allocated time.

Possible Solution: Similarly to above, if cleaning takes too long, smaller datasets can be cleaned and used for initial stages while more data is cleaned simultaneously for later.

- Due to the complexity of the task, the model may fail to outperform the baseline model even after sufficient training.

  Possible Solution: The scope of the project may be limited, for instance to images with only a single car as the prominent object. If this doesn't remedy the issue, reasons for this result can be explored in the progress report to understand possible design flaws, allowing for model improvement before the final implementation.

6 Things to Learn

In order to proceed, we must learn how to:

- Manipulate and clean the COCO data. COCO provides an API, but we must learn how to use it, and determine if it is enough for our intended cleaning.
- Implement the proposed architecture in PyTorch, since it is different from those learned so far. A key difference is the usage of intermediate layers of a pre-trained network, and feeding them to a custom network's layers.
- Implement a baseline model for the task. This may require more research into simpler segmentation methods such as K-means clustering [12]. This is important for debugging and evaluating the neural network's performance on the cleaned dataset.

7 Ethical Issues

The primary ethical concern of our project is intellectual property. Several images contain logos, watermarks and credits of the image owner. Our software could potentially remove these items, and ultimately violate fair usage of the image. Malicious users may even purposely automate the removal of these items using our software.

References

[1] A. Jepson and D. Fleet. (2007). Image segmentation, [Online]. Available: http://www.cs.toronto.edu/~jepson/csc2503/segmentation.pdf. (accessed: 02.18.2019).

[2] M. Smith. (Nov. 2009). American bullfrog in green grass, [Online]. Available: https://www.nationalgeographic.com/photography/photo-of-the-day/2009/11/american-bullfrog-grass-pod/. (accessed: 02.18.2019).

[3] A. Smith and M. Anderson. (Mar. 2018). Social media use in 2018, [Online]. Available: http://www.pewinternet.org/2018/03/01/social-media-use-in-2018/. (accessed: 02.23.2019).

[4] (Dec. 2018). The global autonomous vehicle market is expected to reach $30 billion by 2023: Key analyses forecasts, [Online]. Available: https://www.businesswire.com/news/home/20181227005139/en/Global-Autonomous-Vehicle-Market-Expected-Reach-30. (accessed: 02.23.2019).

[5] (2013). Removing a background from an image in photoshop, [Online]. Available: http://www.microknowledge.com/remove-background-photoshop/. (accessed: 02.24.2019).

[6] J. C. Caicedo, J. Roth, A. Goodman, T. Becker, K. W. Karhohs, M. Broisin, M. Csaba, C. McQuin, S. Singh, F. Theis, and A. E. Carpenter, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images," *bioRxiv*, 2019. DOI: 10.1101/335216. eprint: https://www.biorxiv.org/content/early/2019/02/06/335216.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2019/02/06/335216.

[7] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.

[8] (2018). Coco common objects in context, [Online]. Available: http://cocodataset.org/#home.

[9] G. Shperber. (Aug. 2017). Background removal with deep learning, [Online]. Available: https://towardsdatascience.com/background-removal-with-deep-learning-c4f2104b3157. (accessed: 02.22.2019).

[10] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pretrained on imagenet for image segmentation," *ArXiv e-prints*, 2018. eprint: 1801.05746.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. arXiv: 1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556.

[12] S. Yuheng and Y. Hao, “Image segmentation algorithms overview,” *CoRR*, vol. abs/1707.02051, 2017. arXiv: `1707.02051`. [Online]. Available: `http://arxiv.org/abs/1707.02051`.