# Nice Shirt! - Employing Clothing Detection with Neural Networks to Visually Differentiate Individuals

Ujjaini Das Saarthak Mohan Akshar Shrivats Rohit Shetty

***Abstract*—Visually indexing individuals has many potential applications in the field of human-robot interaction (HRI). Facial recognition may prove beneficial in instances of prolonged, direct human interaction with a personal system or when dealing with sensitive private information; however, many applications of HRI simply involve differentiating a human from other humans rather than naming them individually.**

**In this paper, we aimed to explore methods for differentiating humans through a camera feed using a multi-layered approach for clothing detection. Through our model, we employ the darknet Robot Operating System (ROS) package (utilizing YOLO) to detect, isolate, and publish images of humans given a general camera stream. We then use a region convolutional neural network (RCNN) model, trained with the Clothing Co-Parsing (CCP) dataset, to identify specific articles of clothing per individual that differentiate them from others. The model for detecting human figures is functional. On the other hand, our model for identifying clothing, serving as the second stage in a two-model process, requires further development for adequate testing and training. The final step of comparing the clothes identified by the second stage with people seen prior also needs further development. We anticipate better future results with a more encompassing dataset and the potential use of a model specific to clothing identification.**

***Index Terms*—neural networks, human-robot interaction, clothing identification, darknet, YOLO**

## I. Introduction

Vision recognition serves as a critical component to many modern human-robot interaction (HRI) systems, especially as a tool for differentiating between objects or figures to determine robot actions [1]. Potential systems may be required to distinguish between humans to operate, such as to provide a service, track movement, or categorize subjects.

Facial recognition with the use of convolutional neural networks (CNNs) can be used to perform such classifications. Such is relevant in security-related applications; however, employment of facial recognition technology is stringent on a clear frontal-view of a test subject's face [2]. Furthermore, recognition of a human subject based on exact facial features may not be necessary for all applications, such as the tracking of an individual to store temporary data regarding that person. Distinguishing of a figure based on traceable features, such as the color or type of clothing worn, is sufficient for differentiating between individuals for many use cases.

The application envisioned for this paper involves a robot that patrols the University of Texas at Austin computer labs, identifying and differentiating between various standing and seated individuals. The model would be able to temporarily store information about each person present in the labs, identifying them based off of a feature vector that describes the color and/or style of their clothing. For instance, the model may be able to track for how long an individual has been seated; on the other hand, the robot may be able to distribute amenities, in which case it would need to track which individuals have already received which items. Such a set of robot tasks involves the ability to reliably classify differences between figures, such as through their clothing. It would then utilize these differences to index people as their own category, storing information about that person under their category.

This paper explores an approach for differentiating figures given an image with three stages - a model for isolating the image of a person, another model for producing masks around prominent pieces of clothing worn by the subject, and a final step comparing with people the model has seen before. This idea can be further developed and integrated with a temporary database to store subject information; however, the focus of this paper lies in clothing recognition for distinguishing from other clothing.

## II. Background

Semantic segmentation is an approach for image classification that steps beyond assigning labels to objects within bounding boxes as it additionally incorporates localization. This provides information regarding its spatial location and can be integrated with more dense labeling methods. With semantic segmentation specifically, labels are assigned to each pixel of an image to indicate its class and location, creating a classification that masks image contents more precisely than a bounding box [3]. This can be applied to clothing, as shown in "Fig. 1". Such a concept heavily inspired this paper's approach in utilizing a classification model that generates masks for articles of clothing, discussed further in the Method section.

Deep learning models are common in the field of semantic segmentation; notably, AlexNet and VGG-16 utilize CNNs that have been considered as standard approaches [3].

CNNs, common in the field of image classification, assign weights and biases (that may be trained) to indicate importance to different features of an input image. The structure of a CNN involves the overlapping of various "vision fields" that deconstruct an image. This approach is favorable over a simple

Fig. 1. Clothing detection with semantic segmentation.

feed-forward neural network due to its ability to incorporate the complexity of a two-dimensional image, encapsulating spatial dependencies that are difficult to maintain with a one-dimensional array of pixel values.

### A. AlexNet

AlexNet, presented by Krizhevsky et al. [4], consists of an architecture with five separate convolutional layers. This is a relatively simple approach to a deep CNN.

### B. VGG-16

University of Oxford's Visual Geometry Group (VGG) proposed a deep CNN with 16 weight layers [5]. In VGG-16's architecture, the first weight layers have smaller receptive fields that progressively get larger through all 16 layers, creating a model that is easier to train due to having less parameters [5].

### C. Feature Pyramid Networks (FPNs)

Martinsson and Mogren [6] proposed an approach to semantic segmentation of clothing with CNNs based on an FPN. A CNN is consisted of different pooling layers or stages of specific dimensions. With an FPN, the last features from each stage are are combined to create a pyramid of features (due to their gradually decreasing spatial dimensions) and concatenated to receive a final prediction [6].

These approaches, although effective, relies on a less complex image set with a single individual, such as in "Fig. 1". For the purposes of our paper, using a CNN is beneficial; however, we additionally required a method that was able to handle the complexity of a live video stream.

### D. Combination of Models

Such a need leads to discussion of Ahmad et al. [7] and their work with video surveillance footage. This research study heavily influenced the approach described in our paper as it utilizes a combination of models to extract different aspects of an image from live video. A figure is first detected using a Faster region convolutional neural network (Faster-RCNN), and then the person is tracked with the use of a regression network [7]. Faster-RCNN, proposed by Ren et al. [8], contains a traditional set of convolutional layers; however, it additionally incorporates a region proposal network (RPN) which predicts whether or not an object is present and provides its corresponding bounding box.

## III. Method

Taking into account the various approaches to semantic segmentation and instance classification with CNNs, our team employed a three-step process to first identify individuals in a frame, further segment their appearance based on their clothing, and then finally identify the person from its memory of people it has seen before. This is done through a pipeline that takes a live video stream from the onboard Azure Kinect webcam that would be mounted on a Building-Wide Intelligence (BWI) robot. "Fig. 2" contains a visualization of our identification process.

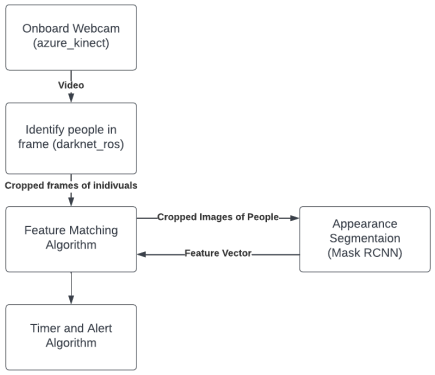


Fig. 2. Image Processing Pipeline

### A. Instance Segmentation

To identify people, an implementation of the YOLOv3 model [9] was used, specifically because it provides greater performance which is critical for running on real-time video footage as input. Additionally, YOLOv3 yields bounding boxes as the output; this is essential for the following step, which divides the image into figures according to their bounding boxes to simplify analysis for the slower Mask R-CNN model. We used the pre-trained weights (trained on the Microsoft COCO dataset [10]) and used the bounding boxes for recognized people. The implementation utilized was darknet_ros [11], an adaptation of the darknet implementation of YOLOv3 that takes input and output from Robot Operating

System (ROS) topics. The camera footage from the Azure Kinect camera was published to a ROS topic, which gets read by darknet_ros. darknet_ros then publishes the bounding boxes to a different ROS topic (see "Fig. 3"). The output of a classified image of a figure is then published as another ROS topic for other packages or the BWI robot to subscribe to.

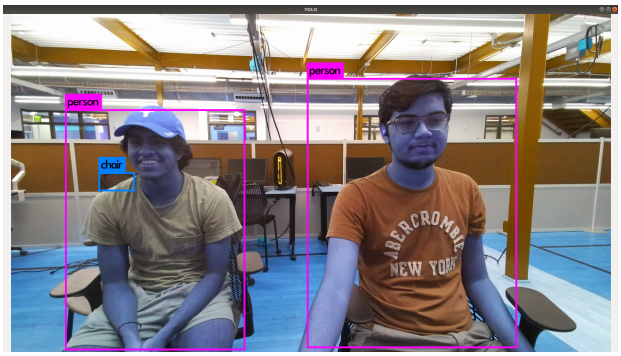


Fig. 3. Bounding boxes on people in webcam footage

### B. Semantic Fashion Segmentation

The following step would have been to identify the clothing in the given image of a figure. Bounding boxes, although useful for the first step, would be inefficient for clothing identification due to the precision needed for understanding clothing boundaries and to analyze clothing color. To more closely match the results of semantic segmentation, our team employed the use of masks rather than bounding boxes. We chose Mask R-CNN for ROS [12], since despite being slower than YOLOv3, the model yields pixel-level masks. This ROS package did not contain software for training new models; therefore, the main Mask R-CNN implementation [13] was used to train a new model. We tested numerous datasets to optimize feature diversity and type of image. These datasets are described further in detail below. The feature vector of clothing (along with the color of each article of clothing) is then to be sent to the final step on the pipeline.

*1) Fashion-MNIST:* Fashion-MNIST [17] is a dataset consisting of 28×28 grayscale images of 70,000 fashion products from 10 categories, with 7,000 images per category. The training set has 60,000 images and the test set has 10,000 images. Fashion-MNIST shares the same image size, data format and the structure of training and testing splits with the original MNIST. We found this dataset to be the most widely used in training neural networks and is even included in the Keras.datasets package. However, due to the low resolution and grayscale nature of the set we noticed poor results when testing the model on our raw input frames.

*2) Fashionista:* Fashionista [18] was a unique fashion dataset as it not only provided annotations for clothing but also for pose estimation and body posture. We thought that this would be an advantage during the training process as we would be able to determine if an individual was sitting down or standing using body posture context. However, the added features of the dataset resulted in poor results due to the lack of overall depth in fashion diversity and choice.

*3) DeepFashion/DeepFashion2:* A large-scale clothes database, which has several appealing properties: 1) DeepFashion contains over 800,000 diverse fashion images ranging from well-posed shop images to unconstrained consumer photos. 2) DeepFashion is annotated with rich information about clothing items. Each image in this dataset is labeled with 50 categories, 1,000 descriptive attributes, bounding box and clothing landmarks. 3) DeepFashion contains over 300,000 cross-pose/cross-domain image pairs [16]. DeepFashion2 is a comprehensive fashion dataset. It contains 491K diverse images of 13 popular clothing categories from both commercial shopping stores and consumers. It additionally contains 801K clothing items, where each item in an image is labeled with scale, occlusion, zoom-in, viewpoint, category, style, bounding box, dense landmarks and per-pixel mask. Moreover, there are 873K Commercial-Consumer clothes pairs. The dataset is split into a training set (391K images), a validation set (34k images), and a test set (67k images) [15]. The predominant issue we ran into when using these datasets were that the annotation methods they used were incompatible with our R-CNN training harness, resulting in the inability to begin training.

*4) Clothing Co-Parsing Dataset:* The Clothing Co-Parsing (CCP) dataset was an elaborately annotated set containing 2000+ high resolution street fastion images with 59 different tags for tops, bottoms, and accessories [14]. It included a wide range of items and provided pixel level mask annotations for over 1000 images in the dataset as well. This seemed like the most practical dataset for our application as it has many images in a similar format to what we would expect to give our R-CNN from the YOLO model.

### C. Recalling from Memory

Once a feature vector is received from the Mask R-CNN network, we required incorporation of knowledge of people we have seen already. Since individuals typically change clothes every day, there cannot be a preexisting database of clothes that correspond with people (in the same way that faces have preexisting databases). Therefore, we must build up such a database ourselves, and that database can only last for a day at most. When we receive a feature vector, we see if there is an approximate match of features in the database. If so, it is likely that the person we see now is the same person as the person we saw previously. If there is no match with anyone, then it's possible that we have not seen this person before, so they will be recorded in the database. The robot will build up a working memory of people that it has seen before; when it sees a person again, it will be able to recall from its memory if that person has already been seen.

## IV. Evaluation

With our current implementation, we isolated 700 images from the CCP dataset for training and 304 images for verification. Had we been able to successfully complete the training

process with the Mask-RCNN model, we would divide the training process into several epochs. At the end of epochs, images from the verification would be used to track the model's cost (inaccuracy) and how it is improving during training.

Furthermore, following our training process, we would use two main metrics to determine the performance of our model - precision and recall. Precision is a measure of the correctness of classes that are identified in an image. Recall, on the other hand, measures whether classes are picked up by the model at all. Maximizing both of these metrics would serve to optimize our model.

## V. Results

We succeeded in completing stage 1: bounding boxes of people were sent to ROS topics correctly. Work on stage 2 is in progress, as we were unable to complete the issues surrounding training the neural network. We did not reach stage 3 due to it being dependent on the still incomplete stage 2.

### A. Problems Faced

In training the second fashion-detection model, we encountered a variety of problems that led to us not completing in time. The abundance of varied datasets led to many early prototypes that were too specialized to a particular category of images and did not generalize to the often occluded photos we would receive from the BWI-robot. This was particularly a challenge with datasets such as the Fashion-MNIST and Fashionista. Additionally, limited experience with GPU acceleration resulted in inefficiency during our model training process with the AHG robotics laboratory machines. Combined with a lack of personal compute power, this meant new model creation was time intensive. To combat this issue, many models were developed with a limited subset of the overall data and thus starved in variety of apparel choices. Furthermore, there dependency-related complications. Mask R-CNN, written with Python, contained a broad requirement listed in its requirements.txt, and a later version of a dependency would have breaking changes. Due to Python being an interpreted language, these errors do not present themselves at compile time, but rather at runtime, and the errors do not clearly indicate that the issue had to do with an upgraded version. This magnified the time needed to produce a functional model.

### B. Limitations

There are some problems that may be inherent limitations of our solution that we will further discuss. There will often be times when bounding boxes of people will overlap, such as when people are close by or when limbs are outstretched. This could be a problem when clothes from the other person get incorrectly attributed to the clothes on the current person and could potentially cause the database to be filled with misinformation. Some heuristic may be needed to guess whether clothes are on the same person. A related problem is clothes in the environment. This risk should theoretically be minimized because the bounding box trims it down to the person, but this may still pose incorrectness when the bounding box is large (because of outstretched limbs) and if the environment is in a place with clothes in the environment (such as a clothing store).

Another problem is if people with similar clothing (such as a sports team) are all in a location simultaneously. All of these people will be misidentified as the same person. Perhaps combining this approach with another identification method (like hair color or skin color) will serve to mitigate this risk.

At times, the network misidentifies parts of the footage as people when there is no person there. This should be rare, but could cause problems if clothes are mistakenly identified as well.

## VI. Discussion

As outlined with aforementioned limitations, our approaches to clothing identification inspired by semantic segmentation require further development.

The most predominant area of improvement is regarding our dataset; although proving to be the most compatible with the tools used in our project, the CCP dataset is limited. It does not offer a wide variety of sitting and standing postures which are likely to be encountered in HRI situations. Furthermore, our first model does not encompass images that show figures that overlap or interrupt each other's bounding boxes. Further research in finding sufficient datasets is required; furthermore, given the complications we faced in finding datasets, taking the images ourselves is an option. However, this would require additional resources to both capture and label hundreds to thousands of training and verification images.

Finally, our tests may be further developed to indicate system effectiveness against other models. For instance, our semantic segmentation model may be tested against a facial recognition software for the same purpose, comparing accuracy, safety, and overall efficiency of the systems.

## VII. Conclusion

Overall, our system addressed the problem of needing to identify and differentiate individuals without clear access to a view of their face to employ facial recognition. Our solution involves using external aspects, such as the color of the person's clothing and possibly their hair and skin color in the future to differentiate them from other individuals. This is done through a three-stage pipeline. First, bounding boxes are creating around individual people in a live video stream. Second, these isolated images of individual figures are sent to a Mask-RCNN model to segment the color of aspects of their appearance. Finally, these results are referenced to an internal database to create a probability that this person has already been seen, indexing individuals as different from others. With this paper, we have successfully accomplished step one.

## References

[1] M. Shridhar and D. Hsu, "Interactive Visual Grounding of Referring Expressions for Human-Robot Interaction," arXiv:1806.03831, 2018.

[2] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah and S. U. Ali, "Facial Recognition using Convolutional Neural Networks and Implementation on Smart Glasses," 2019 International Conference on Information Science and Communication Technology (ICISCT), 2019, pp. 1-6, doi: 10.1109/CISCT.2019.8777442.

[3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," Applied Soft Computing, Volume 70, 2018, p. 41-65, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2018.05.018.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] J. Martinsson and O. Mogren, "Semantic Segmentation of Fashion Images Using Feature Pyramid Networks," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 3133-3136, doi: 10.1109/ICCVW.2019.00382.

[7] M. Ahmad, I. Ahmed, F. Khan, F. Qayum, and H. Aljuaid, "Convolutional neural network–based person tracking using overhead views," International Journal of Distributed Sensor Networks, June 2020, doi:10.1177/1550147720934738.

[8] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv:1506.01497, 2016.

[9] Redmon, Joseph and Farhadi, Ali, "YOLOv3: An Incremental Improvement", arXiv, 2018.

[10] Lin, Tsung-Yi, et al. "Microsoft Coco: Common Objects in Context." Computer Vision – ECCV 2014, 2014, pp. 740–755., https://doi.org/10.1007/978-3-319-10602-1_48.

[11] Bjelonic, Marko. 'YOLO ROS: Real-Time Object Detection for ROS'. N.p., 2016–2018. Web.

[12] Ochiai Akio, The ROS Package of Mask R-CNN for Object Detection and Segmentation, 2017, GitHub repository, https://github.com/akio/mask_rcnn_ros

[13] Abdulla, Waleed. 'Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow'. GitHub repository 2017. Web.

[14] Yang, Wei and Luo, Ping and Lin, Liang. 'Clothing Co-Parsing by Joint Image Segmentation and Labeling'. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2013. Print.

[15] Yuying Ge and Ruimao Zhang and Lingyun Wu and Xiaogang Wang and Xiaoou Tang and Ping Luo,DeepFashion2 - A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images, (CVPR), 2019

[16] Liu, Ziwei and Luo, Ping and Qiu, Shi and Wang, Xiaogang and Tang, Xiaoou, DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[17] Ochiai Akio, Fashion MNIST, 2018, Kaggle, https://www.kaggle.com/datasets/zalando-research/fashionmnist.

[18] G. Rhodes, Fashionista Dataset, 2017, GitHub repository, https://github.com/grahamar/fashion_dataset.