International Journal of Electronic Commerce Studies
Vol.13, No.1, pp.69-92, 2022
doi: 10.7903/ijecs.1953

Single Stage Deep Transfer Learning Model for Apparel Detection and Classification for E-Commerce

Ssvr Kumar Addagarla
Vellore Institute of Technology (VIT)
ssvrkumar@gmail.com

Anthoniraj Amalanathan
Vellore Institute of Technology (VIT)
aanthoniraj@vit.ac.in

ABSTRACT

Although many computer vision-based object detection techniques have evolved in the past decade, it suffers from inconsistent detection accuracy, especially for multiclass classification problems. This paper proposed an approach using the Single Stage Deep Transfer Learning model (SS-DTLM) for multiclass apparel detection using a customized YoloV3 algorithm by adapting 3-level Spatial pyramid pooling (SPP), a multi-scale image feature extractor for faster and reasonable apparel detection and classification. This approach produced a reasonable Mean Average Precision (mAP), reliable object detection, and classification. Our model trained and tested on Open Images Dataset (OIDV4) with six object classes and Custom built Apparel Dataset with five object classes of apparels. Finally, experimental results compared with baseline Yolov3 and Yolov3-Tiny algorithms. Further, this paper also emphasized the detected image's various color spaces using SS-DTLM by applying the K-Means clustering algorithm for further analysis.

Keywords: Custom Object Detection, Yolov3, Spatial pyramid pooling, Color Space, Apparel detection

1. INTRODUCTION

Computer vision is a phenomenal research area, especially for face recognition[1, 2] nowadays. The computer vision approaches also extend to the other research areas, including Social networks and the e-commerce field [3] for identification, recognition, and classification of the products from the vast unclassified datasets[4,5]. The need for object detection and recognition is essential nowadays for growing electronic commerce in developing countries like India, China, Japan, etc. Object detections help e-commerce for various categories of products to better serve the people in their day-to-day activities. With the lots of products listing in multiple categories especially for

apparels ranging from Male, Female, Children, Age and in each type there are several major sub-classifications like Top wear (Ex Shits, T-shirts, Kurta, etc.) and bottom wear (Ex: Pants, Skirts, etc.). As there is a variety of various sub-categories of the products, it is complex to search and retrieve using the limited metadata features of the product. Each product listed on the website may not have all meta information. Earlier image-based retrievals are the basis of the manual feature extraction by applying the computer vision based methods like Histogram of oriented Gradients (HOG) and Local binary patterns (LBP) are to recognize the object in the images[6,7]. In the recent developments in Convolutional Neural Networks (CNN), using deep learning models works on large-scale unstructured data like images, videos, etc., to extract all of its features and self-processed to detect and recognize the necessary objects in the images and videos[8-12].

1.1 Background

Artificial Neural Networks (ANN) [13,14] are widely popular for several decades for the many real-life applications which ability to learn intricate relationships in-between inputs and outputs which are non-linear and complex, especially for image processing, character recognition, forecasting, etc. ANN consists of three essential layers, which are input, hidden, and output layers. ANN uses various functions such as one-hot encoding, softmax for the multiclass classification problem, cross-entropy to minimize the error rate and dropout, batch normalization, L1 and L2 normalizations are using to overcome the overfitting problem in the ANNs [15].

The concept of Deep learning is widely popular for many areas of applications since the mid 2000s. Unlike ANN, Deep learning (a subset of machine learning) methods are self-teaching and learning upon understanding the information provided through various hidden layers. Several deep knowledge-based network frameworks such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Recursive Neural Networks, Long short term memory (LSTM) is applying for various real-time tasks including classification, detection, segmentation, localization, image captioning, similarity learning and video analysis [9,16]. Different methods applied in the deep learning frameworks are backpropagation, stochastic gradient descent, learning rate decay (adaptive learning rate), dropout, max pooling, batch normalization, etc. Several popular pre-trained CNN Architectures, including AlexNet, VGG Net, Google Inception, Xception, ResNet, Yolo, etc. All these architectures are vastly well trained on the large-scale multiclass datasets on multiple machines for several days. Many of the industries and researchers are using these pre-trained models for their purpose. In the given Table 1, we have presented a comparison list of various CNN based network architectures.

2. SIGNIFICANCE OF RESEARCH

The E-commerce industry is growing rapidly, especially in Asian countries like India, China, Sri Lanka, etc., for the past decade [25]. Online fashion has wide popularity

among the people in these countries and around the world due to the availability of more fashion products. To get the desired fashion product from the large-scale databases from the fashion portals like Amazon India, Flipkart, Myntra, etc., are using the various orthodox artificial intelligence-based content and collaborative filter search and retrieval methods [26, 27]. There is a necessity to simplify user actions on the e-commerce portals to get the desired similar and other relevant products based on the selected product image by detecting and recognizing the various attributes in the image.

Table 1. Comparisons of Various CNN based Architectures in terms of Year of Introduced, Input size, Layers, and Parameters

CNN Architecture Type	Year	Input size	Layers	Parameters	Remarks
LeNet-5 [17]	1998	32×32×1	5	60K	LetNet becomes typical architecture and motivation for many others
AlexNet [18]	2012	224×224×3	8	60M	Rectified Linear Units (ReLUs) was implemented first as Activation function
VGG-16 [19]	2014	224×224×3	16	138M	Designed for the deeper networks
Inception-v1 [20]	2014	224×224×3	22	5M	Used dense module/block to build the network
Inception-v3 [21]	2015	299×299×3	42	24M	Incorporated batch normalization
ResNet-50 [22]	2015	224×224×3	50	26M	Uses skip connections and batch normalization
Xception [23]	2016	299×299×3	36	23M	Uses depthwise separable convolutional layers based on CNN
ResNeXt-50 [24]	2017	224×224×3	50	25M	Increases number of parallel paths in each module compare to ResNet-50

3. RELATED WORK

Paul Vioal & Michael Jones presented rapid object detection performed using boosted cascade on the MIT+CMU frontal face data set using a new image representation method called an integral image and a learning algorithm based on AdaBoost. Further, the authors achieved frontal face object detection accuracy with fewer computations [28]. Leaf recognition algorithm was proposed by Stephen Gang Wu et al. using probabilistic neural networks. In this approach, authors used to train on the 1800 leaves,

which classifies 32 various kinds of plants by extracting the 12 different feature from the leaves and later orthogonalized into 5 principal variables to train and test the model and achieved 90% accuracy on the majority of the species and some of them are shortfall [29].

Digitalization of the old handwritten documents is essential nowadays to ease access to the users, and recognizing the content in the documents is essential in the computer vision arena. In [30], an offline hand recognition model is proposed using multidimensional recurrent neural networks (MRNN). For generalized text recognition for the given raw input pixel data and irrespective of the language can be applied and utilized the Neural network Architectures like MRNN, Long Short Term Memory (LSTM), Connectionist temporal classification and network hierarchical structure. Tested their model using IFN/ENIT database, which consists of 32,492 images of Tunisian town names and achieved accuracy of 91.4%.

Neural network-based fish recognition extracted the 18 distinct features of 20 different fish families, utilized 350 fish images for training and testing, and achieved 86% accuracy [31]. For this modeling process, the authors adopted a Multilayer feed-forward neural network with a backpropagation model. E-commerce gains considerable popularity since the Mid 2000s, and there is a necessity for text and image classification with relevant attributes for better recommendations to the users. Lukas Bossard et al. proposed apparel classification with styles and developed a benchmark dataset of over 80,000 images for their apparel classification [32] and adopted Random Forest, Transductive Support Vector Machine (TSVM), and Transfer Forest for their classification task. Image classification using Deep Neural Networks (DNN) is widely developing, and Christian Szegedy has developed precise object localization with DetectorNet. Using Convolutional DNN and performing their experiments on the PASCAL Visual Object Challenge (VOC) 2012 Dataset consisting of approximately 11K images, they later compared their results with several existing models [33].

Brian Lao and Karthik Jagadeesh developed an improved fashion classification along with clothing retrieval using Region-based Convolutional Neural Networks (R-CNN) trained on the Apparel classification with styles (ACS) dataset and colorful-fashion (CF) dataset [34]. They have developed and trained the model on top of AlexNet Convolutional Network and used the weights of the pre-trained ImageNet.

A region proposal network has been proposed by the Shaoqing Ren et al., using Faster R-CNN to improving the object accuracy using localization. Further, VGG-16 and ZF-Net as the base models and trained on the Pascal VOC 2007, VOC 2012 datasets. Which further calculated the mean Average Precision (mAP) for their object detection [35]. Kaiming He et al., has experimented using deep residual learning for the image recognition on the ImageNet classification dataset and training performed especially on Plain Networks (inspired from VGG Network) with 18 and 32 layers, Residual networks (ResNets) with 18, 34, 50, 101 and 152 layers for identification of better accuracy in image recognition. Out of which ResNets have better optimization than Plain Networks, and finally, authors conclude that ResNets with 152 layers have better

accuracy than all other network models. Authors also performed their method on the popularly known Pascal VOC and MS COCO data sets for object detection [36].

Ju-chin Chen and Chao-Feng Liu have developed three distributed computing level Architectures for the visual-based recommended clothing using deep learning. They have experimented on the public clothing datasets and attempted to produce retrieved top clothing based on the given user query [37].

Warinthorn Kiadtikornthaweeyot, Adrian R.L. Tatnall are proposed for the Region of interest detection using histogram segmentation, and the method can be applied for the forest, agriculture, urban areas. In this process, they have taken satellite images to train and test histogram segmentation, morphology dilation, and image ROI on the given datasets [38]. In [39], the authors attempted to classify a few apparels classes using convolution neural networks based on the GoogleNet Architecture. They collected 5093 images of 5 different classes of data from the fashion e-commerce portal Myntra. In [4], the authors have developed a model to detect the apparel by considering the pose and its priors. In detecting the apparel, they used R-CNN as an object detector and considered localization, size, and aspect ratios of the fashion alongside the final posterior detection using the Support Vector Machine (SVM). Fashionista dataset used for the experimentation and all images are annotated at the image pixel level and produced.

Han Xiao, Kashif Rasul, and Roland Vollgraf have developed a new benchmark dataset called Fashion-MNIST, consisting of 60,000 grayscaled images with ten different fashion categories of equal size. They have experimented with their dataset on various algorithms like Decision Tree Classifier, GaussianNB, LinearSVC, Random Forest, etc., for five times by shuffling the training data to get better test accuracy and compared results with existing benchmark dataset MNIST [5].

Alexander Schindler, Thomas Lidy et al., had experimented on five different pre-trained architectures (VGG16, VGG19, InceptionV3, Custom CNN, VGG-like) for fashion classification using CNN. Their process collected 7833 images from the various e-commerce portals and performed a 3-fold cross-evaluation and computed accuracy [40]. Detecting human age and gender categorization, a hybrid model named CNN-ELM (Extreme learning machine) was developed by Mingxing Duan, Kenli Li (2018). CNN and ELM are used for the feature extraction and produce an intermediate result for classification applied on the two popular datasets (MORPH-II, Adience Benchmark) and compared with other studies in terms of accuracy and efficiency [41].

Chandadevi Giri, Sheenam Jain et al., had reviewed how artificial intelligence impacted the fashion and apparel industry since 1980 and found most researchers/organizations contributed their work in the fashion industry from 2009 onwards. For this, they have surveyed 149 research articles from Scopus and Web of Science database and concluded that most authors have talked about predictive techniques like regression and SVM, and few talked about Big Data applicability in the fashion industry [42].

Yian Seo and Kyung-Shik shin are proposed a hierarchical CNN (H-CNN) model for the apparel classification. The authors have used the popular VggNets (Vgg16 and Vgg19) as the base models, and newly constructed H-CNN trained and tested on the Benchmark Fashion MNIST dataset, which generates the results loss and accuracy for to solve the multi-classification error in fashion datasets [11].

4. RESEARCH METHOD

In this computer vision and deep learning arena, objection detection on the images plays a vital role in lowering the human intervention to detect and process the various objects for the many e-commerce websites. Several object detection algorithms based on convolutional neural networks are already in place to detect multiple objects, especially two-stage detectors and Single-stage detectors. Two-stage detectors are R-CNN [43], Fast RCNN [44], Faster RCNN [45], Mask R-CNN[46], etc., initially introduces region sets by region proposal network or selective search and then classifier applied. In the Single-shot detectors like SSD[47], Yolov1 [48], Yolov2(Yolo9000) [49], Yolov3 [50], etc., can predict with the help of bounding boxes and take down the region proposals from the process to predict final class probabilities. Thus, Single-stage detectors are faster enough in detection compared to two-stage detectors, whereas single-stage detectors are not far behind in case of accuracy. In the given Table 2, we illustrated the various model detection with Mean Average Precision (mAP).

Table 2. Various object detection results generated using MS COCO and PASCAL VOC datasets.

Model	Backbone Architecture	mAP @50
Fast R-CNN [44]	VGG-16	35.9
Faster R-CNN [45]	VGG-16	42.7
R-FCN [51]	ResNet-101	54.8
Faster R-CNN+++ [36]	ResNet-101-C4	55.7
Mask R-CNN [46]	ResNeXt-101	62.3
SSD300 [47]	VGG16	43.1
YOLOv2 [49]	Darknet-53	57.9
YOLOV3 [50]	Darknet-19	44.0
SSD512 [47]	VGG16	48.5
SSD321 [52]	ResNet-101	45.4

4.1 Transfer Learning

In Machine learning, the concept of Transfer Learning (TL) is beneficial to train a new model for a specific task from the existing model by taking some part of the parameters to achieve the outcome for the model. In this context, let's say a Model M_a is trained on a large generic dataset D_a to achieve a result for T_a's given task. However, for another model M_b, utilizing the dataset D_b is to achieve an outcome and preventing efficient model training for a different task T_b. In this case, we use some partial or full

existing pre-trained model parameters to speed up the current model to achieve efficient results for the different tasks. We can implement the Transfer Learning tasks for a similar domain with other tasks. Especially for a particular task, if we can have a limited dataset to train a new model, the TL approach is useful to transfer the knowledge from the pre-trained model to the new model to speed up the process. Even for a social cause, the TL approach minimizes the Graphics Processing Unit's total running time (GPU) or Tensor Processing Unit (TPU). It reduces the CO2 effect on the environment generates from large cloud platforms.

4.2 Deep Transfer Learning

Further, applying the TL approaches in Deep Learning, often referred to as Deep Transfer Learning Techniques. The top-level view of the TL approach is shown in Figure 1. There are several ways of using the TL in Deep learning, depending on the domain-specific task. For many deep learning tasks, extraction of the Features from the images and videos are more important at the initial level. The Deep TL approach is to apply the Pre-Trained model knowledge and weights for the initial feature extractor only but not to update the final model weights and final fully connected layers for a new task.

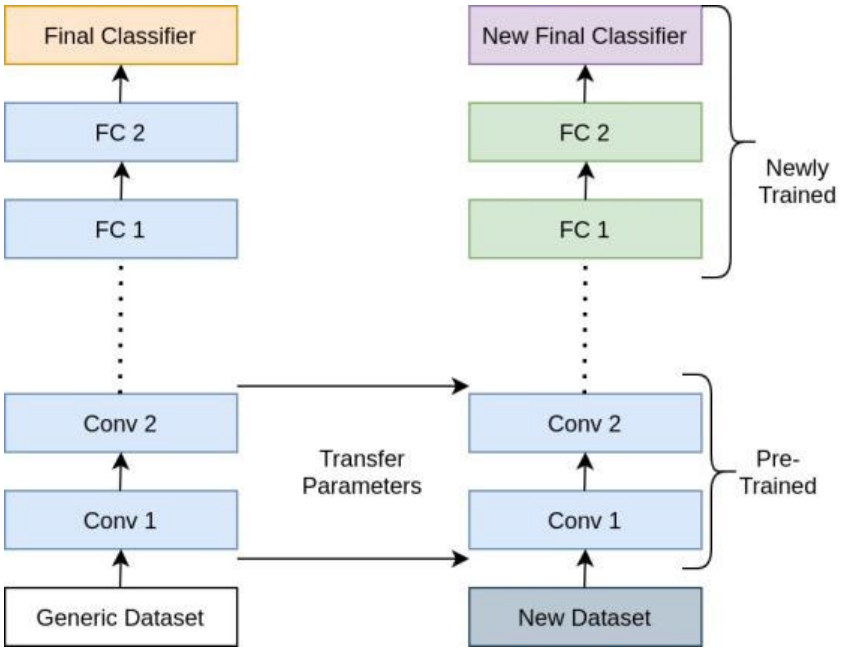

Figure 1. Top-Level Diagram for Deep Transfer Learning Approach

4.3 Single Stage Deep Transfer Learning

Object detection has an immense role in many areas of applications, including e-commerce platforms. To achieve the object detections either in images or videos, two-stage object detectors, and Single-stage object detection are two effective use of Transfer learning approaches. In Two-stage object detection, apart from the initial feature extractor, the model initially performed identifying the prospective regions and knew it as Region of Interests (RoIs) in the first level. Then in the second level, the

model tries to downsample the recognized ROIs to identify the requisite objects using bounding boxes.

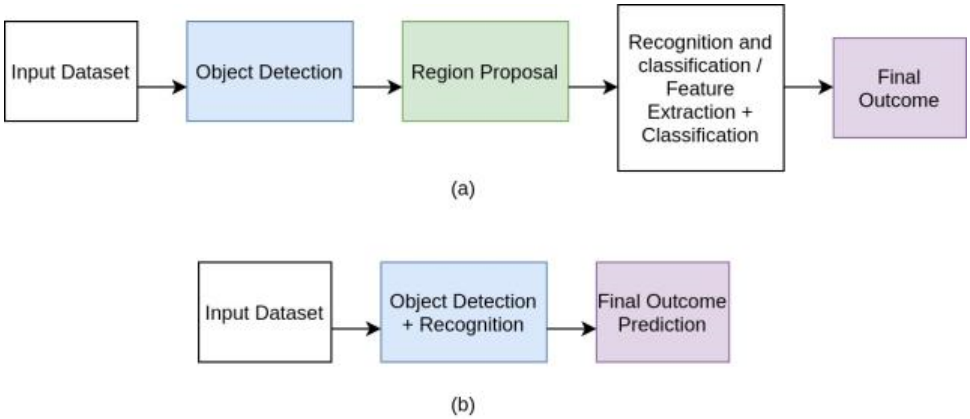

Figure 2. Top Level View of (a) Two-stage object detector (b) Single-stage object detector

The Single-stage object detector treats the object detection as a simple regression problem, predicts the multi-bounding boxes for the objects, and computes the final classification task's probabilities. Here in this approach, the model utilizes the Intersection over Union (IoU) approach for appropriate bounding box prediction for the objects when training the classification task. The Model view of the Single-stage and Two-stage object detector are shown in Figure 2.

In our Single-Stage Deep Transfer Learning Model (SS-DTLM), we have adapted Customized Yolov3, Yolov3 with Spatial Pyramid Pooling (SPP) [53, 54] and Yolov3-Tiny [55, 56] architectures. Our method utilizes the Darkent-53[50] as a backbone network, which is faster enough and less complex architecture and produces a reasonable accuracy score for the object detection. Figure 3 describes the structure of our proposed Methodology.

4.4 Yolo Models

Detecting the objects and recognition in the images or videos is a heuristic task. Several researchers have proposed various classification and detection algorithms. In which Yolo (You only look once) based models (Yolov1, Yolov2, Yolov3, etc.) are faster and accurate in detecting and predicting the objects in the given images or videos. In our work, we incorporated Spatial Pyramid Pooling (SPP) with Yolov3 Model by adapting the GIou metric instead of IoU for better object predictions. In Table 3, we have presented a comparison between the Yolo family of Models and our proposed model.

Table 3. Model comparison between various Yolo versions and the proposed method.

Parameter	Yolo	Yolov2	Yolov3	Yolov3-SPP (Our Method)
Network Architecture	Yolo Customized (Inspired from GoogleNet Model)	Darknet-19	Darknet-53	Darknet-53
Input Grid Cell size	7x7	13x13	13x13	13x13
Batch Normalization	No	Yes	Yes	Yes
Anchor Boxes Prediction	No	Yes	Yes	Yes
Fine-Grained Feature Detection	No	Yes	Yes, Improved	Yes, Improved
Multi-Scale Image Training with various resolutions	No	Yes	Yes, Improved	Yes, Improved
Classifier	Softmax	Softmax	Binary cross-entropy with Logistic activation	Binary cross-entropy with Logistic activation
Metrics Used	Precision, Recall, mAP	IoU, Precision, Recall, mAP	IoU, Precision, Recall, mAP	GIoU, Precision, Recall, mAP
Remarks	High localization error, detects only 49 objects, and only two boxes are predicted by each grid cell.	It is improved in detecting smaller objects with faster speed.	It Introduced objectiveness score and Feature Pyramid Networks for faster and better object detection.	It Utilizes 3-Level max pool sizes & GIoU for better and faster object detection & prediction.

4.5 Yolov3

As a baseline model, Yolov3 divides the given input image into $S \times S$ (13×13) (shown in Figure 4) grid cell and each of the grid cell primary constraints to predict the bounding boxes B and probability of class C *of objects.* If the object has been detected by multiple grid cells in the given image, and this detection problem is achieved by applying the Non-Max Suppression (NMS) [57, 58]. NMS initially computes the Intersect Over Union (IoU) [59], where the area of the intersection region dived over the union of two bounding boxes.

$$Intersection\ over\ Union\ (IoU) = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (1)$$

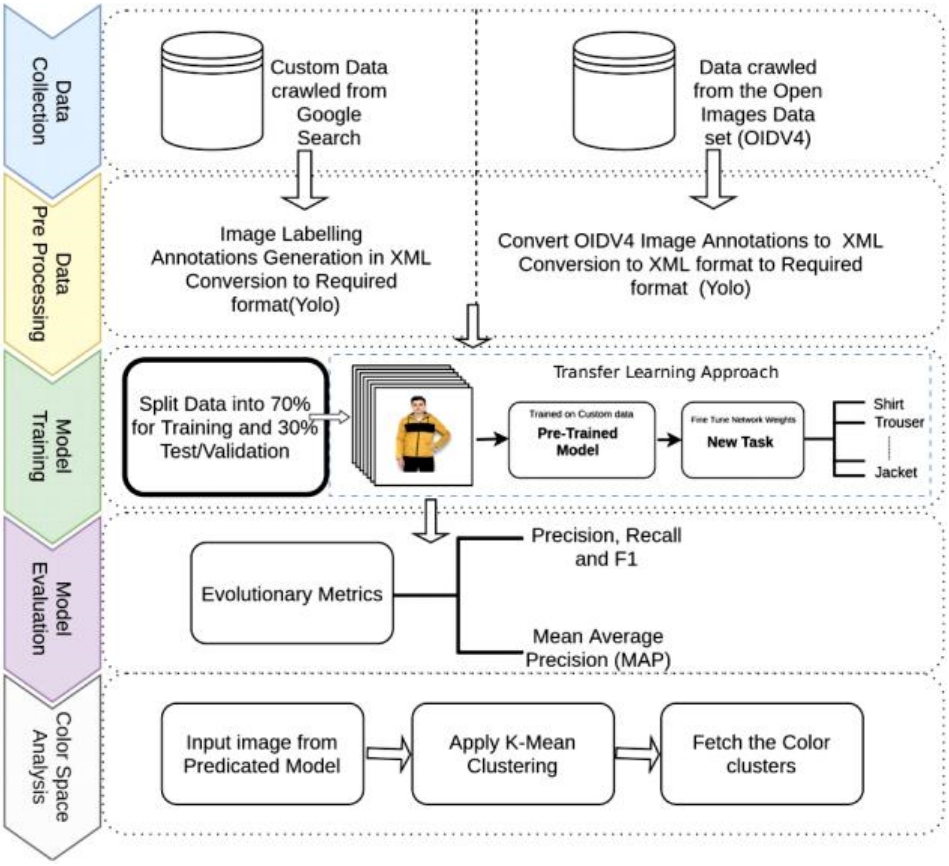

Figure 3. Proposed Structure of the image-based Object Detection using SS-DTLM

Whereas the IOU has infeasible to optimize the non-overlapping bonding box in some cases and to overcome this weakness, we adapted a new metric called Generalized IoU (GIoU) [60]. The GIoU can be computed as follows.

$$Generalized\ IoU\ (GIoU) = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \quad (2)$$

In GIoU, consider the smallest box denoted as C that encloses the ground truth bounding box. Considering GIoU threshold set to a maximum of 0.5 and a bounding box greater than 0.5 are excluded from the process due to higher IOU of the corresponding object, which helps detect proper bounding boxes for object detection and continues to find all the objects. Yolo uses three different scales of anchor boxes for every scale detection. Total 9 anchor boxes are ranging as {(10×13), (16×30), (33×23)}, {(30×61), (62×45), (59×119)}, {(116×90), (156×198), (373×326)} are used at first, second and third scale and tensor is S × S × [3 × (4+1+C)] for the four bounding box coordinates (t_x, t_y, t_w, t_h), one objectness score (P_0) and C class Predictions $(P_0, P_1 \ldots\ldots P_c)$.

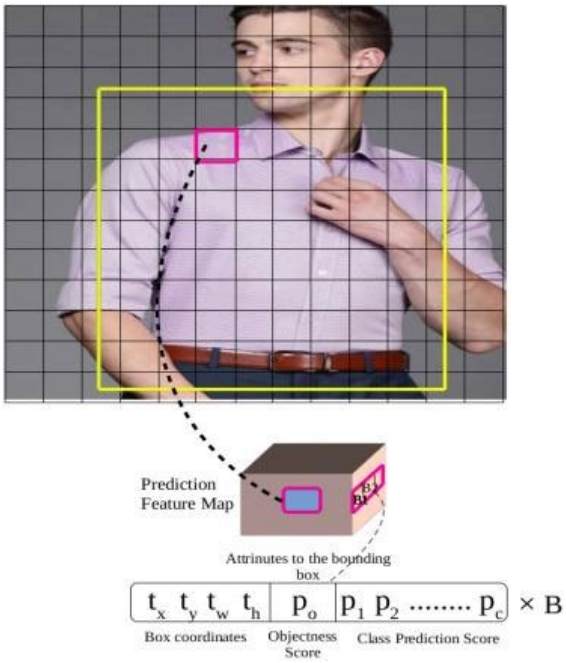

Figure 4. A 13x13 Image grid for detecting the Shirt

Feature Extractor:

Earlier versions of Yolo use the Darknet-19 as backbone feature extractor architecture, which is low at detecting small objects. In YoloV3, we adapted the new network architecture called Darknet-53 shown in Figure 5 [50], which has 53 convolutional layers and robust than ResNet-101 and ResNet-152.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 5. Darknet-53 Network Architecture

Darknet-53 has shortcut connections with 3x3 and 1x1 convolutional filters. Yolov3 uses the Feature Pyramid Networks (FPN) [61], and three predictions are made. This

process further performs the upsampling technique from earlier layers to get semantic and fine-grained information from previous feature maps to process and improve the final output. Figure 6 illustrates the full architecture of the Yolov3 Model.

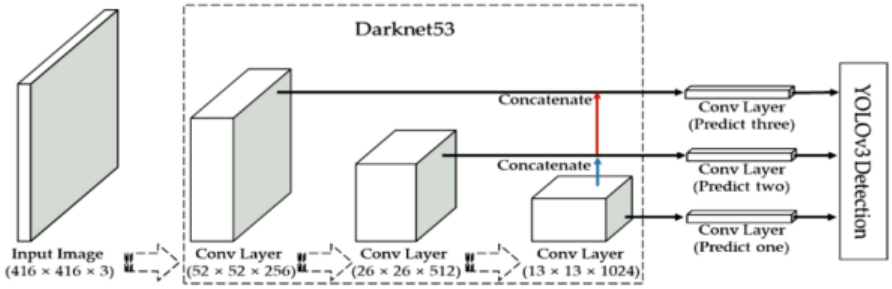

Figure 6. Full Architecture of the YoloV3 [62]

Class Predictions:

Compared with the earlier versions of the Yolo family of algorithms, whereas the softmax (last layer) is used to predict the object's binary class. Nowadays, it's necessary to predict the multiclass labels, especially when dealing with Open Images data set [63], which are multi labeled. Considering the above case, we have replaced the softmax with the logistic linear regression classifiers (model illustration shown in Figure 7), responsible for producing the multiple object predictions in lieu with IoU and binary cross-entropy loss function is used during the training.

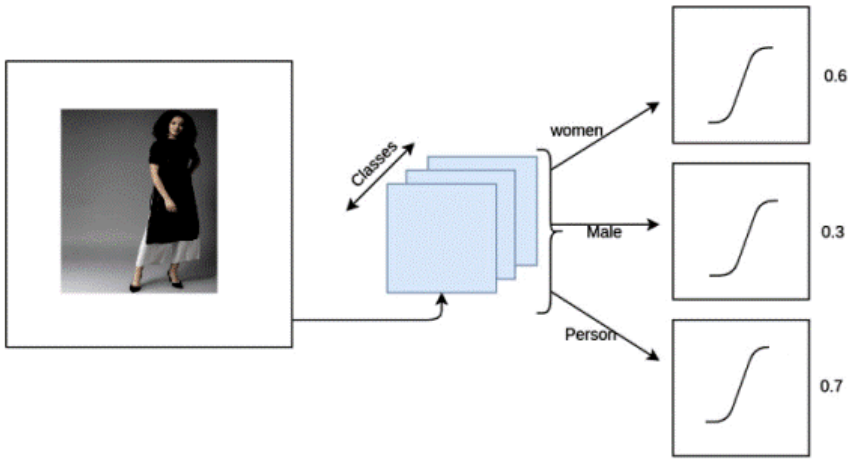

Figure 7. YoloV3 Interpretation for Multiclass Prediction for a given input image

Loss Function:

The model loss function is used to understand how our model is trained or learned on the given network architecture and other hyperparameters. The trained model predicts the correct bounding boxes for the corresponding classes. Loss function [48] in YoloV3 computes in 4 stages and which are centroid loss (x_i, y_i), width, height loss (w_i, h_i), objectness loss (obj, noobj)which computes objectness score 0 or 1 and finally computes the classification loss. The following formula computes all four of these loss functionalities.

$$
\begin{aligned}
&\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{j=0}^{S^2} 1_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
\tag{3}
$$

4.6 Yolov3-SPP

Spatial Pyramid pooling/matching [53] method can help detect multi-scale images more efficiently when dealing exclusively with the deep learning algorithms for the classification task, which improves the performance and accuracy of the objects trained on convolutional neural networks. A fixed dimensional input is expected from the fully connected layers when training/testing with multi-scale images to our model. Spatial pyramid pooling is an efficient approach to train the network of images of multiple scales.

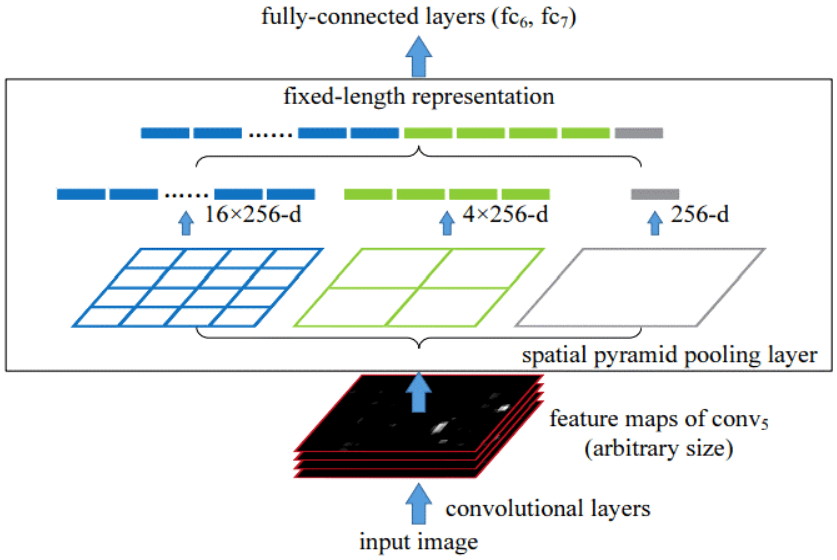

Figure 8. 3-Level Spatial Pyramid pooling layer

Our proposed SS-DTLM approach incorporated SPP into the YoloV3 for better performance and SPP performs Max pooling on the Yolov3 networks last convolutional or sub-sampling layer. Figure 8 [53] shows the 3-level SPP, which computes a dimensional vector of N×B where N is the filters in convolutional layers, and B is the number of Bins that is a constant value. In the next step, it fed into the fully connected (FC) layer. In this 3-level SPP, initially, pooling is done on each feature map. Thus, a 256 dimension vector is created and creates 4 values of 256 dimension vector, 16 values of 256 dimension vectors finally forms 1 dimension vector by concatenating all these three vectors. In the last level, the concatenated 1-d vector is further fed into the FC

layer. So, we can avoid crop and resizing the images to fixed sizes like any other CNN architectures.

4.7 Yolov3-Tiny

Compare to Yolov3 and Yolov3-spp model, Yolo-Tiny is a lighter version that has a less complex architecture with only nine 1x1 and 3x3 convolution layers and Max pooling for the feature extraction. The tiny version of YoloV3 is faster in detection than other full specification models, which is low at mean average precision (mAP) for the given objects. As in the YoloV3 network, it also uses shortcut connections for extracting the better features and applies the linear logistic regression for the classification task at the last layer.

4.8 Color Space Analysis

Clustering is an unsupervised learning technique to make similar types of objects grouped into various clusters. After classification is done in our apparel detection model, we aim to analyze the colors in the detection model's output image. For this process, we have implemented K-Mean clustering in our approach.

Initially, we fed the input colored image, and further image normalization applies to smoothening. The process involves resizing the shapes and color conversions from BGR(Blue, Green, Red) to RGB (Red, Green, Blue). Then we needed to initiate and get the HEX values of the image and initiates the K Mean clustering, which randomly picks the center points and applied Euclidean distance measure between points in the image color space. We will then find the nearest values and frame the cluster and repeat it until the required condition meets. Then we arrange all the generated HEX values of the color into ascending order along with the percentage of cluster colors.

Algorithm for Color Space Analysis

Input: Color Image
Output: Color Clusters with a percentage of various color distribution

```
Read input Image
calculate the img shapes
colors.Normalize(vmin=-1.,vmax=1.)
RGB2HEX(color)
Resize the image
procedure(get-color)
    do compute HEX values
    KMeans(n_clusters = number_of_colors)
    do Calculate Euclidean Distance Measure between color points
    Repeat the process until terminate condition satisfied
    sorted(counts.items())
    plot the HEX values of the clustering colors
end procedure
```

4.9 Data Collection

We collected two different apparel datasets for experimentation and analysis to address the object detection and classification on mid-large and small object samples. For the first approach, we have downloaded six object classes of each 2K images of Shirts, Trousers, Jeans, Skirt, Dress and Jacket a total of 6K images from Google Open Image Dataset (OIDV4). OIDV4 dataset consists of 600 object classes of around 9 million images and pre-annotated of various sizes. For the second process of our object detection on small samples, we have manually crawled five object classes of each 100 images of shirts, trousers, t-shirts, sarees, and women-kurtas from the Indian e-commerce fashion websites of various sizes and sample collection shown in Figure 9 for both the datasets.

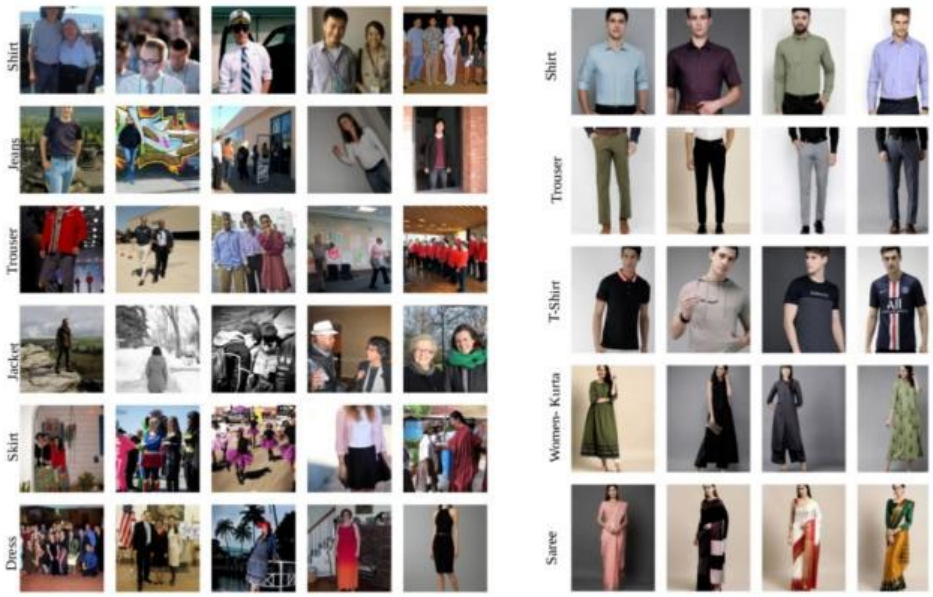

Figure 9. Sample images for 6 class of objects from OIDV4 Dataset (left) and 5 class of objects from Custom built dataset (right)

5. EXPERIMENTATION AND RESULTS

In this experimentation section, we have trained and validated using the transfer learning approach on YoloV3 with pre-trained YoloV3 weights, YoloV3-SPP with pre-trained weights of SPP weights, and finally on the Tiny version of Yolo with YoloV3-tiny weights with Darknet53 backbone architecture. We have done experimentation in three stages.

5.1 Experimentation 1

This section has used the OIDV4 dataset, pre-annotated class of objects, and applied data pre-processing to convert into the required format. For pre-processing, first, we have converted object annotations into PASCAL VOC format, then we have transformed into the required Yolo format, which consists of each class label and corresponding bounding box information ($x_{min}, y_{min}, x_{max}, y_{max}$). We then split into our data as train values with 70% and test values with 30% of data. We made our custom

object configuration file with various hyperparameters listed in Table 4 for all the 3 YoloV3 algorithms to feed into the network. Further, we have initiated data normalization and augmentation for smoothing the process. We have done the training process on our desktop with a Core i7-8700k processor with 16GB of RAM, 8GB of RTX 2070 GPU, Ubuntu 18.04 operating system. Model Training for YoloV3 and YoloV3-SPP took around 8 hours of time to train the network model. For YoloV3-tiny, it took around 3 hours of time to train the network of 110 epochs and evaluated all three models.

Table 4. Various SS-DTLM YoloV3 hyper-parameters used for experimentation

Parameter Name	Supporting Value
GIoU Loss gain	3.54
Cls loss gain	37.4
Obj Loss gain	64.3
Lr0 (initial learning rate)	0.00579
Momentum	0.937
Weight_decay	0.000484
Degree (image rotation)	1.98
Image translate	0.05
hsv_h, hsv_s, hsv_v (image augmentation)	0.0138, 0.678, 0.36
Nms-thres	0.5

5.2 Experimentation 2

In this section, we have small sample collections of objects crawled from the Indian e-commerce websites. As a significant task, first, we manually labeled all the 500 images of 5 object classes using the LabelImg annotation tool, which converts PASCAL VOC XML format. Then we have converted all the annotated VOC XML files into Yolo format, which we have done as in the earlier section. Then we have split our data into train values with 80% and test values with 20%, respectively. Here also we performed the normalization and data augmentation for smoothening the process. To train and test the model, we have used YoloV3, and YoloV3-SPP pre-trained models with hyper-parameters specified in Table 4 and made other custom configuration figuration files according to the input data, and fed all these details into the network. For running these two algorithms, it takes around 4 hours to train the network, and we have finally tested and evaluated our models.

5.3 Experimentation 3

Finally, we have taken the trained model's output image (YoloV3-SPP) and gives it as an input for the color space analysis. Here in this process, we used Open-CV libraries and methods to normalize the image and then extracted the required HEX values of the image's colors. Finally, applied K-Means clustering by taking the random K cluster centroids further computes Euclidean distance measure of the similarity among the

color space. We produced the results with various colors presented in the input image along with the percentage of domination.

5.4 Results

We have computed and evaluated various performance metrics consists of Precision, Recall, F-measure, and Mean Average Precision (mAP) for the best model on both datasets. Here these metrics are computed confusion matrix as multiclass classification problem. Finally, we have tested and produced different results by taking the input image size as 416×416 and 608×608 with the IoU threshold of 0.5.

Table 5 mentioned the comparison results, and YoloV3-SPP performed well with an mAP of 42.4% on the OIDV4 dataset, and it produces faster and good object detection on the given input images. Surprisingly, on a custom dataset, we performed the annotations manually, where both the YoloV3 and SPP models have got mAP of more than 95%, and the recall percentage is 100%. Table 6 and Table 7 presented the Average precision (AP) of the individual object classes on both the OIDV4 and Custom built datasets. Figure 10 shown final detections done on proposed SS-DTLM using SPP on both the datasets. We have plotted various performance metrics and parameters like GIoU, objectness, and classification during the model training for all the algorithms we have considered for the apparel object detection and classification. Further, we have plotted various training parameters for object detection using YoloV3-SPP to understand the model's performance, which we considered our best-trained model in Figure 11.

Table 5. Comparison of our trained classification algorithms

Trained Model	Dataset	MAP@0.5 (%)	Recall (%)	Image Input size
Yolo-V3	OIDV4	41.4	83.9	416
Yolo-V3-SPP	OIDV4	42.4	85.4	416
Yolo-V3-Tiny	OIDV4	33.5	84.5	416
Yolo-V3	Custom Dataset	97.3	100	416
Yolo-V3-SPP	Custom Dataset	98.4	100	416

Table 6. Average precision of the object classes on OIDV4 Dataset

Trained Model	Shirt (AP)	Trouser (AP)	Jacket (AP)	Skirt (AP)	Jeans (AP)	Dress (AP)	mAP
Yolo-V3	47.6	26.7	38.3	62.7	32.7	40.6	41.4
Yolo-V3-SPP	50.5	27.0	45.0	58.7	33.3	40.2	42.4
Yolo-V3-Tiny	40.3	21.8	34.0	47.3	26.6	31.0	33.5

Table 7. Average precision of the object classes on Custom-built dataset

Trained Model	Shirt (AP)	Trouser (AP)	T-shirt (AP)	Saree (AP)	Women-kurta (AP)	mAP
Yolo-V3	98.9	97.0	97.3	99.5	99.1	97.3
Yolo-V3-SPP	93.3	99.4	98.6	99.5	95.5	98.4

Figure 10. Model Detection Trained on Custom Dataset (left) and OIDV4 Trained Dataset (right)

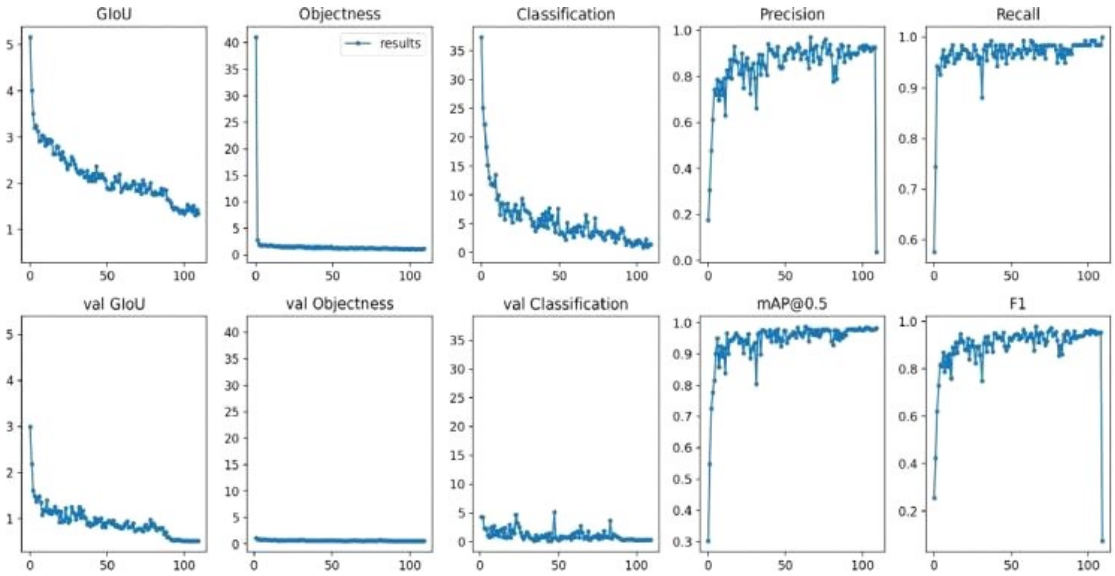

Figure 11. Classification and other performance metrics on YoloV3-SPP using OIDV4 Dataset

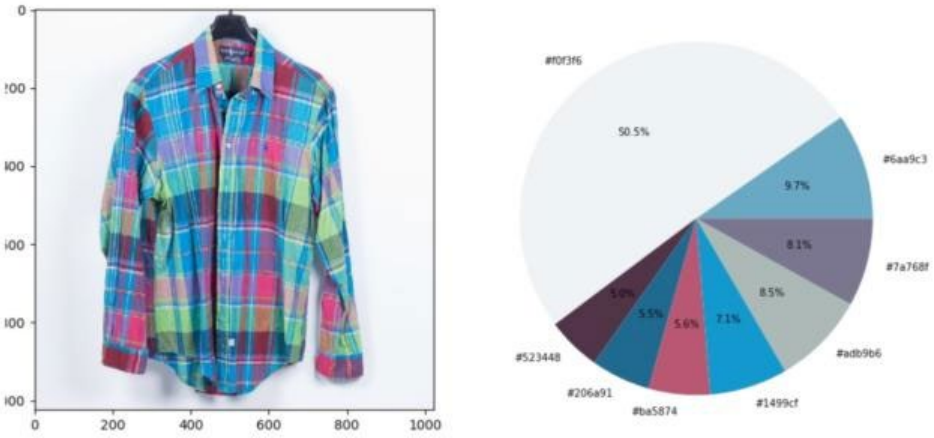

Figure 12. Color Distribution for Input Image (left) and Colors distribution obtained for a given image using K-Means Clustering (right)

Finally, we have analyzed the input image for the color space in RGB (Red, Blue, Green). We got the final cluster distribution with the percentage of each color dominates in the given input image using K-Means clustering, shown in Figure 12.

6. CONCLUSION AND FUTURE WORK

In our Single Stage Deep Transfer Learning Model (SS-DTLM) apparel detection, we had adapted YoloV3, YoloV3-SPP, and YoloV3-Tiny with the Darknet-53 as the backbone architecture. In our approach, we have implemented a 3-level Spatial Pyramid Pooling in YoloV3 for better object detection without cropping and resizing in the image for Multi-scale local regions for better feature extraction. We further replaced the softmax layer with an Independent logistic linear regression to tackle the multiclass predictions. We further adopted cross-entropy for the object classification loss and vanishing gradient issue instead of mean squared error. The experiments on the OIDV4 apparels dataset and Custom built dataset demonstrates that YoloV3-SPP is more accurate than YoloV3 and YoloV3-Tiny. We also illustrate the color space analysis using computer vision methods and K-Means clustering, which produces fair color cluster distribution for the given input image. Still, the model also considered background colors, even into consideration.

Further, in our future research, we try to eliminate the background color space in the image which is inadequate for correct color distribution and considers. The only selected object in the image by auto-cropping helps us implement to fetch similar and Complementary colored apparels based on the patterns that will help increase the user level experience to boost e-commerce sales.

7. REFERENCES

[1] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, “Face recognition by humans: Nineteen results all computer vision researchers should know about,” *Proc. IEEE*, vol. 94, no. 11, pp. 1948–1962,

2006. https://doi.org/10.1109/JPROC.2006.884093

[2] S. Z. Li, Handbook of face recognition. London, UK: Springer-Verlag, 2011.

[3] S. Zoghbi, G. Heyman, J. C. Gomez, and M.-F. Moens, "Fashion meets computer vision and nlp at e-commerce search," *Int. J. Comput. Electr. Eng.*, vol. 8, no. 1, pp. 31–43, 2016. https://doi.org/10.17706/IJCEE.2016.8.1.31-43

[4] K. Hara, V. Jagadeesh, and R. Piramuthu, "Fashion apparel detection: The role of deep convolutional neural network and pose-dependent priors," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9. https://doi.org/10.1109/WACV.2016.7477611

[5] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv Prepr. arXiv1708.07747*, 2017.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893. https://doi.org/10.1109/CVPR.2005.177

[7] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *2009 IEEE 12th international conference on computer vision*, 2009, pp. 32–39. https://doi.org/10.1109/ICCV.2009.5459207

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. https://doi.org/10.1038/nature14539

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[10] Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. neural networks Learn. Syst.*, 2019. https://doi.org/10.1109/tnnls.2018.2876865

[11] Y. Seo and K. Shin, "Hierarchical convolutional neural networks for fashion image classification," *Expert Syst. Appl.*, vol. 116, pp. 328–339, 2019. https://doi.org/10.1016/j.eswa.2018.09.022

[12] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *arXiv Prepr. arXiv1901.06032*, 2019.

[13] M. H. Hassoun and others, Fundamentals of artificial neural networks. MIT press, 1995.

[14] H. D. Beale, H. B. Demuth, and M. T. Hagan, "Neural network design," *Pws, Bost.*, 1996.

[15] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018. https://doi.org/10.1016/j.heliyon.2018.e00938

[16] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*, 2017, pp. 1643–1647. https://doi.org/10.1109/ICACCI.2017.8126078

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. https://doi.org/10.1109/5.726791

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. https://doi.org/10.1145/3065386

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[20] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, HI, USA, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), HI, USA, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), HI, USA, 2016.

[23] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.

[24] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated Residual Transformations for Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), HI, USA, 2017, pp. 5987-5995, doi: 10.1109/CVPR.2017.634.

[25] "eCommerce - Asia | Statista Market Forecast," 2020. [Online]. Available: https://www.statista.com/outlook/243/101/ecommerce/asia. [Accessed: 02-Feb-2020].

[26] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender SYSTEMS: State of the art and trends," Recommender Systems Handbook, pp. 73–105, 2010.

[27] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, 2017. https://doi.org/10.1016/j.eswa.2016.09.040

[28] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," *CVPR*, vol. 1, no. 511–518, p. 3, 2001. https://doi.org/10.1109/cvpr.2001.990517

[29] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *2007 IEEE international symposium on signal processing and information technology*, 2007, pp. 11–16. https://doi.org/10.1109/isspit.2007.4458016

[30] A. Graves and J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Advances in Neural*

Information Processing Systems 21, 2008, pp. 545–552.

[31] M. K. Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdeh, "Fish recognition based on robust features extraction from size and shape measurements using neural network," *J. Comput. Sci.*, vol. 6, no. 10, p. 1088, 2010. https://doi.org/10.3844/jcssp.2010.1088.1094

[32] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, "Apparel classification with style," in *Asian conference on computer vision*, Berlin, 2012, pp. 321–335. https://doi.org/10.1007/978-3-642-37447-0_25

[33] C. Szegedy, A. Toshev, and D. Erhan, "Deep Neural Networks for Object Detection," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 2553–2561.

[34] B. Lao and K. Jagadeesh, "Convolutional neural networks for fashion classification and object detection," *CCCV 2015 Comput. Vis.*, pp. 120–129, 2015.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99. https://doi.org/10.1109/tpami.2016.2577031

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, NV, USA, pp. 770–778. https://doi.org/10.1109/cvpr.2016.90

[37] J.-C. Chen and C.-F. Liu, "Visual-based deep learning for clothing from large database," in *Proceedings of the ASE BigData & SocialInformatics*, 2015, p. 42. https://doi.org/10.1145/2818869.2818902

[38] W. Kiadtikornthaweeyot and A. R. L. Tatnall, "Region of interest detection based on histogram segmentation for satellite image," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 41, pp. 249–255, 2016. doi:10.5194/isprs-archives-XLI-B7-249-2016

[39] S. G. Eshwar, A. V Rishikesh, N. A. Charan, V. Umadevi, and others, "Apparel classification using convolutional neural networks," in *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, India, 2016, pp. 1–5. doi:https://doi.org/10.1109/ictbig.2016.7892641

[40] A. Schindler, T. Lidy, S. Karner, and M. Hecker, "Fashion and apparel classification using convolutional neural networks," *arXiv Prepr. arXiv1811.04374*, 2018.

[41] M. Duan, K. Li, C. Yang, and K. Li, "A hybrid deep learning cnn–elm for age and gender classification," *Neurocomputing*, vol. 275, pp. 448–461, 2018. doi:https://doi.org/10.1016/j.neucom.2017.08.062

[42] C. Giri, S. Jain, X. Zeng, and P. Bruniaux, "A detailed review of artificial intelligence applied in the fashion and apparel industry," *IEEE Access*, vol. 7, pp. 95364–95384, 2019. doi:https://doi.org/10.1109/access.2019.2928979

[43] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference

on Computer Vision and Pattern Recognition, 2014, pp. 580-587, doi: 10.1109/CVPR.2014.81.

[44] R. Girshick, "Fast r-cnn," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, 2015. https://doi.org/10.1109/iccv.2015.169

[45] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500. https://doi.org/10.1109/cvpr.2017.634

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. https://doi.org/10.1109/iccv.2017.322

[47] W. Liu, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. https://doi.org/10.1109/cvpr.2016.91

[49] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271. https://doi.org/10.1109/cvpr.2017.690

[50] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv Prepr. arXiv1804.02767*, 2018.

[51] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," *Adv. Neural Inf. Process. Syst.*, pp. 379–387, 2016.

[52] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv Prepr. arXiv1701.06659*, 2017.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015. https://doi.org/10.1109/tpami.2015.2389824

[54] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, p. 0. https://doi.org/10.1109/iccvw.2019.00011

[55] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection," in *2018 15th Conference on Computer and Robot Vision (CRV)*, 2018, pp. 95–101. https://doi.org/10.1109/crv.2018.00023

[56] Z. Yi, S. Yongliang, and Z. Jun, "An improved tiny-yolov3 pedestrian detection algorithm," *Optik (Stuttg).*, vol. 183, pp. 17–23, 2019. https://doi.org/10.1016/j.ijleo.2019.02.038

[57] S. Maji and J. Malik, "Object detection using a max-margin hough transform," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009,

pp. 1038–1045. https://doi.org/10.1109/cvprw.2009.5206693

[58] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2147–2154. https://doi.org/10.1109/cvpr.2014.276

[59] A. Rosebrock, "Intersection over Union (IoU) for object detection," *Diambil kembali dari PYImageSearch* [Online]. Available: https//www. pyimagesearch. com/2016/11/07/intersection-over-union-iou-for-object-detection, 2016.

[60] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666. https://doi.org/10.1109/cvpr.2019.00075

[61] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125. https://doi.org/10.1109/cvpr.2017.106

[62] H. Ma, Y. Liu, Y. Ren, and J. Yu, "Detection of collapsed buildings in post-earthquake remote sensing images based on the improved yolov3," *Remote Sens.*, vol. 12, no. 1, p. 44, 2020. https://doi.org/10.3390/rs12010044

[63] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.