National University of Singapore

CS4243: Computer Vision and Pattern Recognition

Where's Waldo: Using color mask, HOG and DCT feature extraction techniques for detection

Guide: *Prof. Angela Yao* ***Students:*** *Aniket Bhatia, Lee Chi Cheng Daniel, Jia Yilin*

Abstract

This project is about detecting Waldo and his friends (Wenda & Wizard) in a range of scanned images from the popular book series "Where's Waldo", where Waldo and his friends are hidden in various colorful and cluttered scenes. This report aims to outline the stages of feature extraction using multiple techniques, namely color mask, HOG and DCT, to detect Waldo and friends. The detection results of the 3 characters are approximately % precision rate, and % recall rate, which indicates that the stages and algorithm detailed in this report does work to some extent, but requires further fine-tuning.

1 Introduction

This project aims to detect Waldo and his friends Wenda and Wizard in a range of scanned images from the popular book series "Where's Waldo", where Waldo and his friends are in various colorful and cluttered scenes. Waldo and his friends sport iconic looks (red and white striped tops for Waldo and Wenda, long white beard for Wizard etc.), which are key to finding them within the scenes.

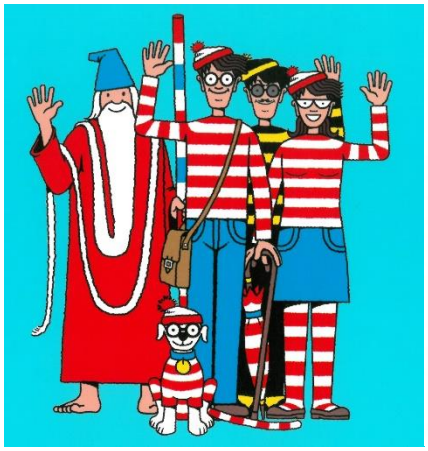


Fig. 1: Waldo and Friends

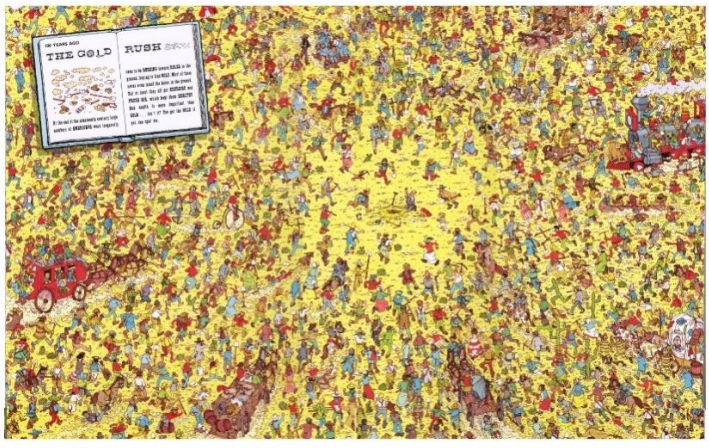


Fig.2: Example Scene to find Waldo and Friends in

As shown in a sample scene in Fig. 2, it consists of many fine features and colors, and also up to hundreds of faces and bodies. In Fig. 1, Waldo, Wenda and Wizard (center, rightmost and leftmost characters respectively) have many iconic features as well, such the striped outfits, blue hat and long beard, shape of glasses etc. Thus, the challenge to detect Waldo and friends cannot be solved

trivially with a single technique as no one technique can take such many features into account and output a perfect single detection for each character, but multiple techniques and fine-tuning over a number of stages to extract and narrow down interest points till an optimal solution is found.

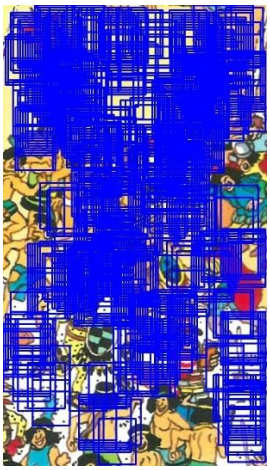


Fig. 3: HOG detection

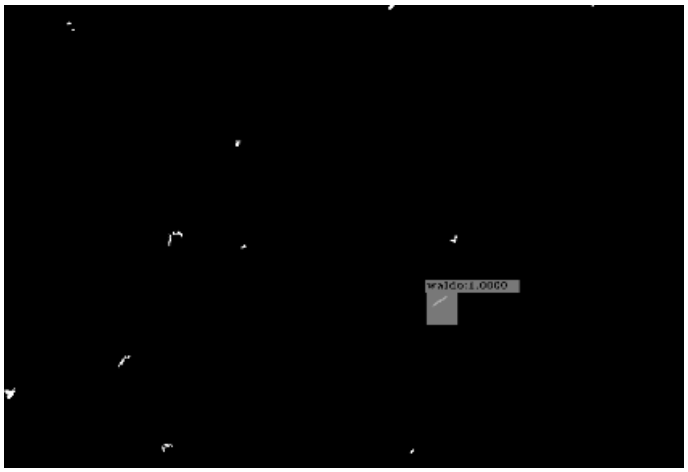


Fig. 4: Color mask on "019.jpg"

In the mid-term report, we detailed standalone experimentations on some feature extraction techniques, namely applying a color mask to image scenes, generating HOG vectors for densely sampled windows, and SURF. Each of these techniques individually fell far short of making a proper detection, such as shown in Fig. 3 where HOG features of densely sampled windows over the cropped image from "000.jpg" yielded so many positive Waldo detections (blue bounding boxes) even after training an SVM with multiple HOG features of Waldo templates as positive samples. This could be due to many random windows having similar HOG features due to the dense and fine features throughout the whole image. While the SURF feature method worked well only when the template was extracted from the image itself and failed when the template was any other image.

One experiment, namely the color mask technique in particular, was effective in surfacing interest points that included Waldo's location (shown in Fig. 4), by only extracting red-colored lines at a certain angle corresponding to the iconic red line on Waldo's hat. This technique involved extracting only red segments and suppressing the rest of the colors, and subsequently suppressing interest points that are too large, or those lines are not within a certain angle range. This works because Waldo and his friends' faces are always oriented in a particular direction (facing toward the right) in all scenes. This technique also can be done fast even on high resolution images (~10000 x 7000 pixels) which are a large proportion of the dataset, as compared to densely sampling windows throughout whole images and transforming them before classifying them.

We thus propose a multi-stage detection algorithm with color mask application technique as the first stage to extract interest points from image scenes. Subsequently we will use HOG to extract feature vectors on windows at the interest points, as well as on positive character templates and negative sampled windows. Finally, we will train an SVM with the positive and negative samples, and use the trained SVM to predict if interest windows are detections or not. This process will be done at multiple scales of window dimensions. This proposed algorithm will be detailed more in the next section.

2 Proposed Solution

Preparation Step: Pre-select Positive and Negative Templates

Before the algorithm is executed, templates were first selected – positive and negative, as we intend to use the templates to train an SVM in a later stage for prediction of extracted interest windows from image scenes.

Positive samples of fixed dimension (50 x 60 pixels) - Waldo, Wenda and Wizard faces templates, and negative scenes –randomly selected cropped image scenes not containing the characters, are prepared and put into folders for later processing. Fig. 5 shows the templates for each character, and Fig. 6 shows some sample scenes without any of the characters below.

Fig. 5: Samples of templates of each character (50 x 60 pixels each)

Fig. 6: Some selected cropped image scenes without any characters

In our implementation, 10 templates for each character are used (Fig. 5 only shows 5 templates for each character), and 14 different cropped scenes without any of the characters from which windows were extracted at a certain step size (Fig. 6 only shows 3 cropped scenes).

Step 1: Define first iteration parameters for Step 3

A certain starting dimension scale is defined for the current iteration with respect to the original template size of all templates in Fig. 5, depending on the selected original image dataset (i.e. 019.jpg) pixel dimensions.

Image dataset pixel dimensions	Starting scale	Starting Template Dimensions
Width or height > 7000	2.5	125 x 150
5000 < width or height <= 7000	2.0	100 x 120
3000 < width or height <= 5000	1.5	75 x 90
0 < width or height <= 3000	1.0	50 x 60

Table 1: Starting scales and dimensions for each dataset image

Step 2: Apply color mask to extract Interest Points

As mentioned in Section 1, we started with color mask application to the original scenes. Fig. 8and 9subsequently shows the interest points from image "019.jpg" generated by applying a color mask as described in the previous section to extract interest points corresponding to the iconic features of Waldo and friends indicated in Fig. 7 below.

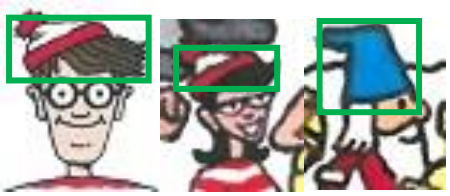


Fig. 7: Iconic features of Waldo and Friends

Fig. 8: Mask of original "019.jpg" image for Waldo and Wenda detection

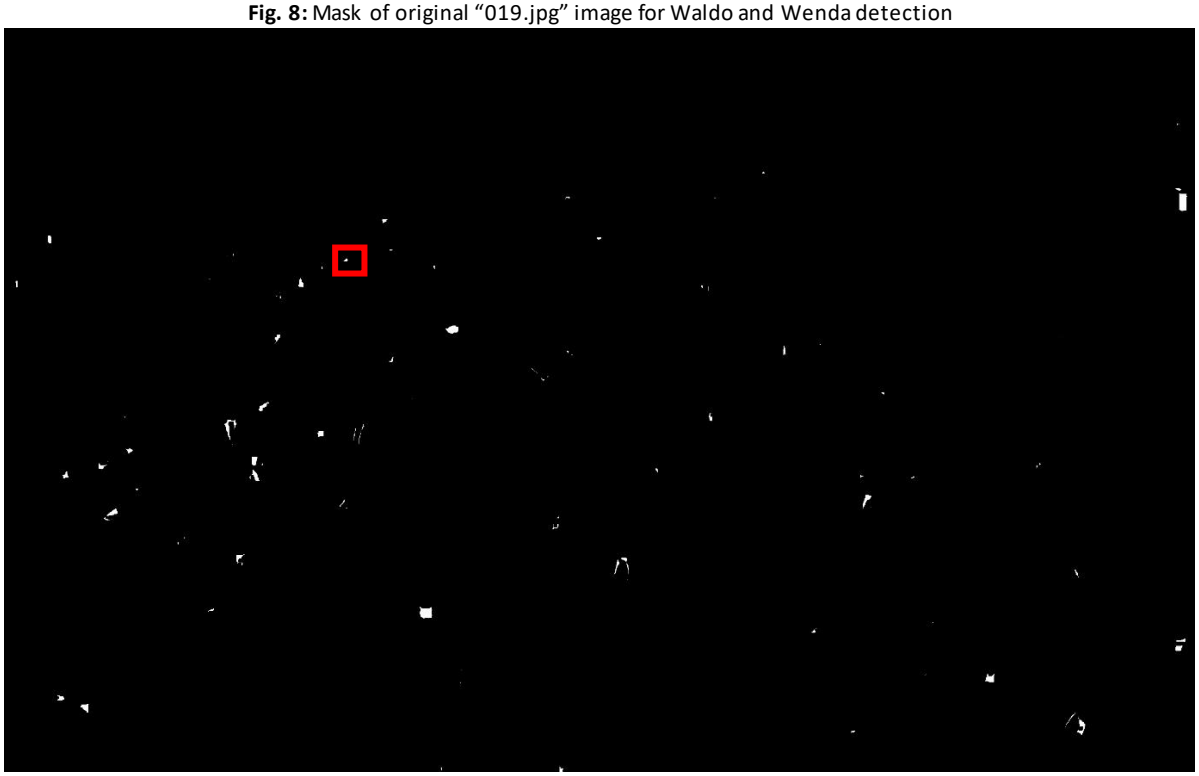


Fig. 9: Mask of original "019.jpg" image for Wizard detection

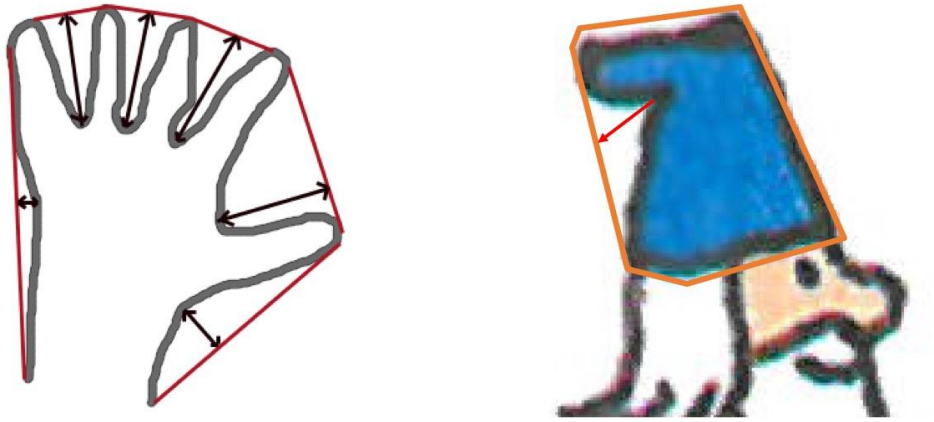
The red bounding boxes show the ground truth for Waldo, Wenda in Fig. 8, and Wizard in Fig. 9. Fig. 8 was generated as described in Section 1 by extracting red line segments corresponding to Waldo's and Wenda's caps, as well as their hair that is also another iconic feature of which the sharp corners are always pointing to the right. The hair detection was an improvement added on after the mid-

term report, and was done similarly to the red line extraction, but this time extracting black segments that correspond to black segments that end with sharp corners and suppressing everything else. The red line and hair are detected with the following steps:

1. Use a color filter to extract regions with specified color range. A binary mask is obtained with this step.
2. Remove noisy spots from the mask.
3. Remove components (the white blobs in the binary mask) with size smaller than the threshold we set.
4. Draw a bounding rectangle around each blob. Obtain the width-to-height ratio of each of the rectangles.
5. Remove blobs with too large/small width-to-height ratio.
6. Fit the longest possible straight line to each blob. Obtain the angle of this line.
7. Remove blob if its angle is out of the range we specified.

After we obtained the masks for red line and hair, we then dilate the extracted black segments and the extracted red line segments and find the overlaps of the two segments, since Waldo and Wenda's hair and cap always go together. The centers of the overlapping areas are our interest points of Wenda and Waldo. Similarly, interest points in Fig. 9 was extracted that corresponds to Wizard's iconic blue cap (blue segments were extracted, the rest suppressed). Since Waldo's and Wenda's iconic features are similar, for every image scene, one set of interest points are generated to be used to detect Waldo and Wenda, and another set to be used to detect Wizard, as Wizard's iconic features are different from Waldo and Wenda. The blue cap of Wizard is detected with the following steps:

1. The first 5 steps are same as above.
6. We found that fitting straight line to wizard cap is a not a good practice here. Instead after step 5, we draw a convex hull around each blobs and count the number of large defects in each blobs. We will discard the blob if the count does not equal to 1. The intuition behind this step is illustrated in the following figure.



The defects to a convex hull are indicated by the arrows in the picture above.

The convex hull of wizard cap has only one large defect, as indicated by the red arrow.

Fig. 10: Explanation of defects of wizard hat

Step 3: Extract Interest windows from original image scene

All templates (Fig. 5) are scaled to according to Table 1 if this is the first iteration for the current dataset image (i.e. 019.jpg), otherwise scaled according to the iteration scale, and all subsequent extracted windows in the rest of the steps are made to be the same dimensions as the scaled templates.

Next, windows from the original image scenes are extracted at the locations of all the generated groups of interest points from the previous step, as shown subsequently in Figs. 10 and 11.

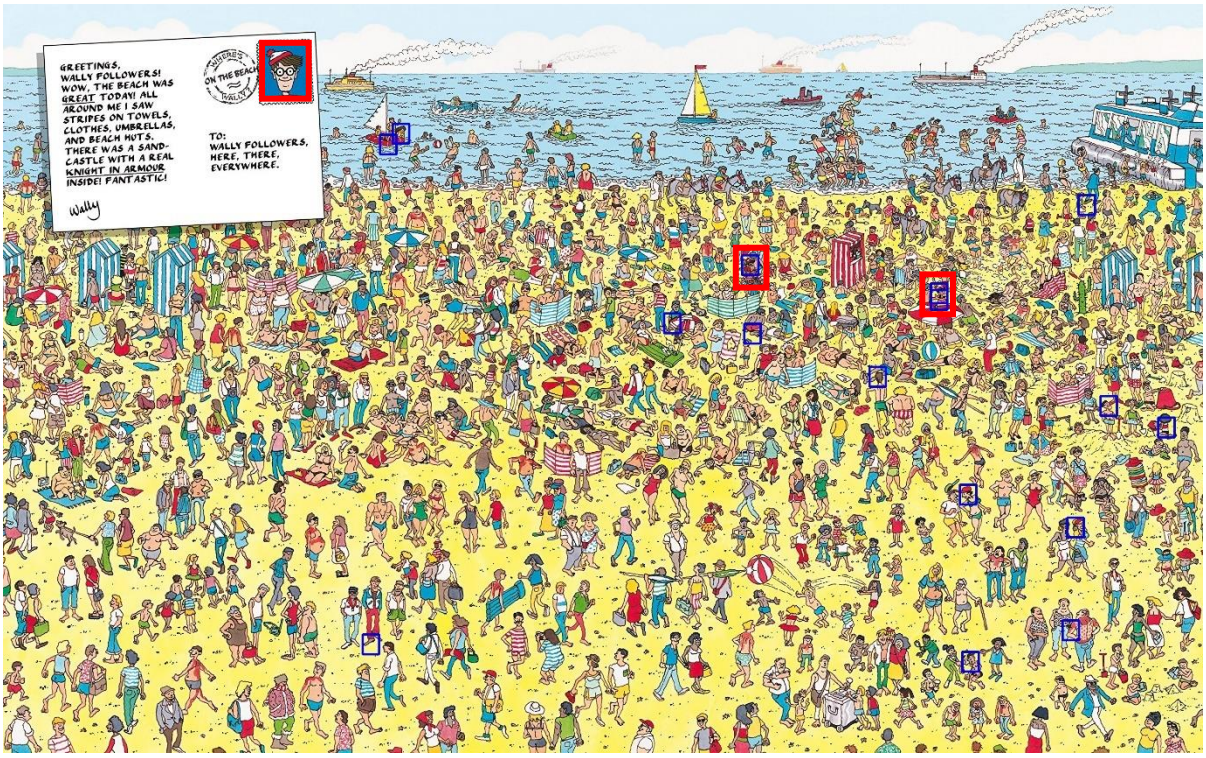


Fig. 11: Windows (Blue bounding boxes) extracted for Waldo and Wenda Detection, corresponding to interest points in Fig.8, red boxes are ground truth for Waldo and Wenda

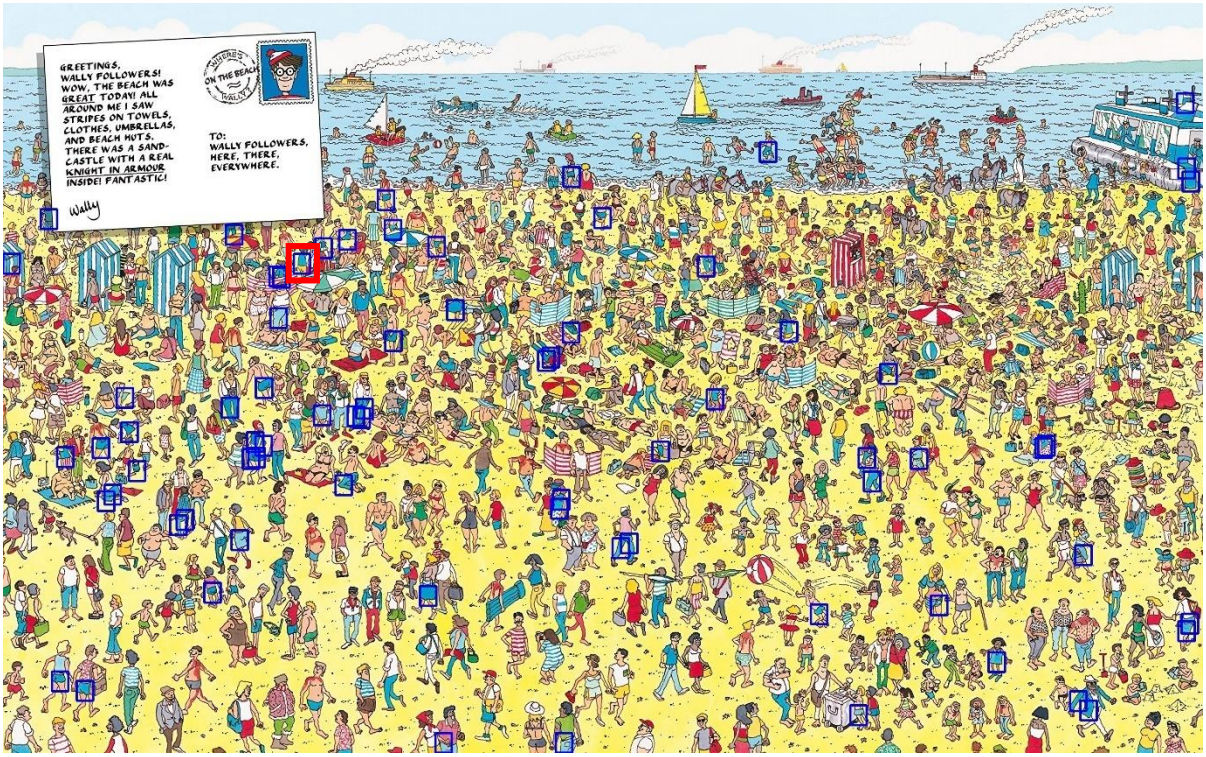


Fig. 12: Windows (Blue bounding boxes) extracted for Wizard Detection, corresponding to interest points in Fig. 9, red box is ground truth for Wizard

Step 4: Extract negative sample windows from negative image scenes

Windows are then sampled at a regular step size from the prepared negative image scenes in the Preparation Step, Fig. 6, as shown in Fig. 12 below.

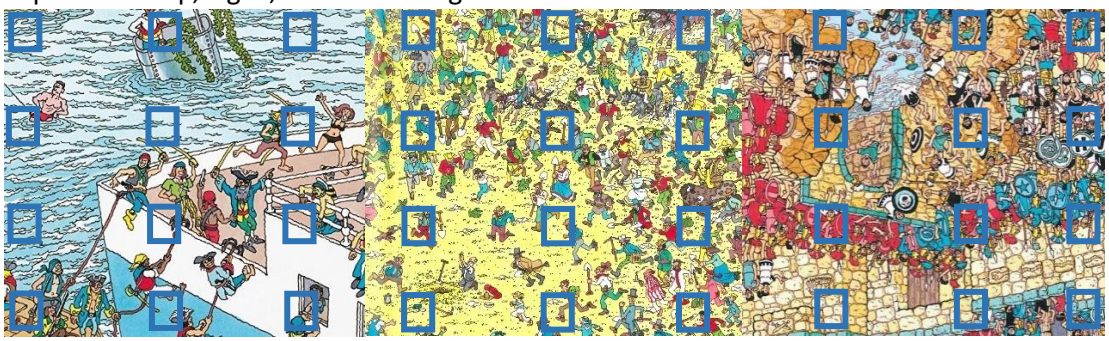


Fig. 13: Samples of windows (blue boxes) sampled from a number of scenes without any of the characters

Step 5: Convert all extracted windows and templates to HOG feature vector representation

All scaled templates, and all windows extracted from steps 3 and 4 are converted into HOG feature vector representation. Since all scaled templates and all extracted windows are of the same pixel dimensions, the feature vector of each templates and extracted windows will be of the same dimensions.

Step 6: Train SVM and Binary Classification of Interest Windows

For each character – Waldo, Wenda and Wizard, an SVM is trained according to the configurations in Table 2 below.

SVM No.	Character	Positive Samples – HOG Feature Vectors of:	Negative Samples – HOG Feature Vectors of:
1	Waldo	Scaled Waldo templates (Fig. 5)	Scaled Wizard templates (Fig. 5) + windows from negative image scenes (Fig. 12)
2	Wenda	Scaled Wenda templates (Fig. 5)	Scaled Wizard templates (Fig. 5) + windows from negative image scenes (Fig. 12)
3	Wizard	Scaled Wizard templates (Fig. 5)	Scaled Wenda templates (Fig. 5) + windows from negative image scenes (Fig. 12)

Table 2: SVM training configuration

From Table 2 above, SVM 1 is then fed with the HOG Feature Vectors of windows in Fig. 10 to get classification of which window(s) are true detections of Waldo. SVMs 2 and 3 similarly are also fed with the HOG Feature Vectors of windows in Figs. 10 and 11 respectively, and used to predict which windows are true detections of Wenda and Wizard respectively. This will give 3 sets of windows in the current dataset image corresponding to detections for each of the 3 characters. If there are positive detections in any of the 3 sets of windows, that respective character's windows will be written to the output txt file for that character in the "output_txt" folder, and an image containing the final bounding boxes of the detections, of original file name with suffix _<"Waldo" or "Wenda" or "Wizard"> will be saved into the "output_img" folder (i.e. 019_waldo.jpg), and the algorithm will not attempt to detect the respective character anymore. If an image has only Waldo and Wizard detected, there will only be 2 files for the current dataset image (i.e. 019_waldo.jpg and 019_wizard.jpg)

A small detail to note in Table 2 is that for SVM 1, we did not use HOG feature vectors from Wenda templates as negative samples, and similarly for SVM 2, we did not use HOG feature vectors from Waldo templates as negative samples. This is because Waldo and Wenda have very similar features, and by including the other character as negative samples, it introduces a lot of ambiguity when the SVM decision boundary is fitted. We found that by not including the other character as negative samples, the detection performance increases significantly.

Step 7: Decrease scale by 0.15, repeat Steps 3 to 6 until all required characters produce at least 1 detection box, or scale drops below 0.4 where HOG feature extraction does not work anymore

Step 8: Repeat Steps 1 to 7 for each dataset image

Summary

Our proposed algorithm to detect Waldo and Friends:

Preparation Step: Pre-select Positive and Negative Templates
- **Step 1:** Define first iteration parameters for Step 3
- **Step 2:** Apply color mask to extract interest points in dataset
  - **Step 3:** Extract Interest windows from original image scene
  - **Step 4:** Extract negative sample windows from negative image scenes
  - **Step 5:** Convert all extracted windows and templates to HOG feature vector representation
  - **Step 6:** Train SVM and Binary Classification of Interest Windows
- **Step 7:** Decrease scale by 0.15, repeat Steps 3 to 6 until all required characters produce at least 1 detection box, or scale drops below 0.4 where HOG feature extraction does not work anymore

Step 8: Repeat Steps 1 to 7 for each dataset image

3 Execution

3.1 Algorithm Fine-Tuning

Parameters we had to fine tune are as follows:

1. Threshold of color segments to extract in Step 1 (blue for wizard, red and black for Waldo's and Wenda's cap and hair respectively
2. Number of each character templates to use (Preparation step, Fig. 5)
3. Number of negative image scenes from which to draw negative sample windows (Preparation step, Fig. 5)
4. Step size for windows to be extracted from negative image scenes (without any characters), as shown in Fig. 12
5. Starting scales for each data set, defined in Table 1
6. SVM training configuration in Table 2

For parameter 1, the color threshold parameter is significant to tune in order to obtain the correct shade of color that corresponds to the iconic features throughout the range of dataset images.

For parameter 2, this could affect the detection accuracy since there are not a lot of character templates to crop out, we tried to take the 10 most representative crops of each character over the whole range of dataset images, such as from high- and low-resolution datasets, from images with varying backgrounds.

For parameters 3 to 4, these affect the training of the SVM. Since we have far more areas in each dataset image that we can draw negative sample windows from than positive character templates for training, if we take negative samples evenly throughout the whole range of dataset images, it could overwhelm the decision area of positive detection due to the huge ratio of negative to positive samples. Hence the negative samples need to be regulated. By cropping out small portion of negative scenes from each of the 70 dataset images, and then choosing 14 of these cropped scenes (parameter 3), and then regulating it further by only drawing windows from each of these 14 scenes at step size of 15 (parameter 4), meaning that we divide each cropped scene in Fig. 12 into the window size and going in row major order, we only sample every 15^{th} window as a negative sample, we are selective about how much negative samples that we use to train the SVMs. We first tried the algorithm using samples at higher frequency over all the 70 cropped negative scenes, which produced very poor detections. Subsequently we gradually reduced the 2 parameters until the above configuration, which gave significant improvement over the initial configuration.

For parameter 5, the easiest way was to start from the highest possible scale found for Waldo and his friends over all the dataset images, for each dataset image. However, since the dataset image resolutions vary quite a lot, from a few hundred to a few thousand pixels for each dimension, it will be very computationally inefficient to iterate through redundantly high scales to detect the characters for low resolution images. Hence to reduce computationally redundant iterations over all dataset images, for each image, depending on the resolution, we define a starting scale that is slightly higher than the scale that the characters face windows are most likely to correspond to, and gradually reduce the scale from there for each dataset image.

For parameter 6, the rationale has been explained on page 8, in the paragraph before Step 7.

Most of the above parameters were chosen arbitrarily with the above described rationales after numerous trial and error. Hence within the time constraints of the project, the above parameter configurations were what we obtained that produced reasonable detection results for each of the characters, which will be described in the following section.

3.2 Results

In this section, we will first display 5 images of positive detections, and 5 images of negative detections, and then provide some statistics on our algorithm after running it on the whole range of 70 dataset images. Detections by our algorithm are indicated with blue bounding boxes, while white shaded boxes are the ground truth annotations from the provided annotations files.

Positive detections:

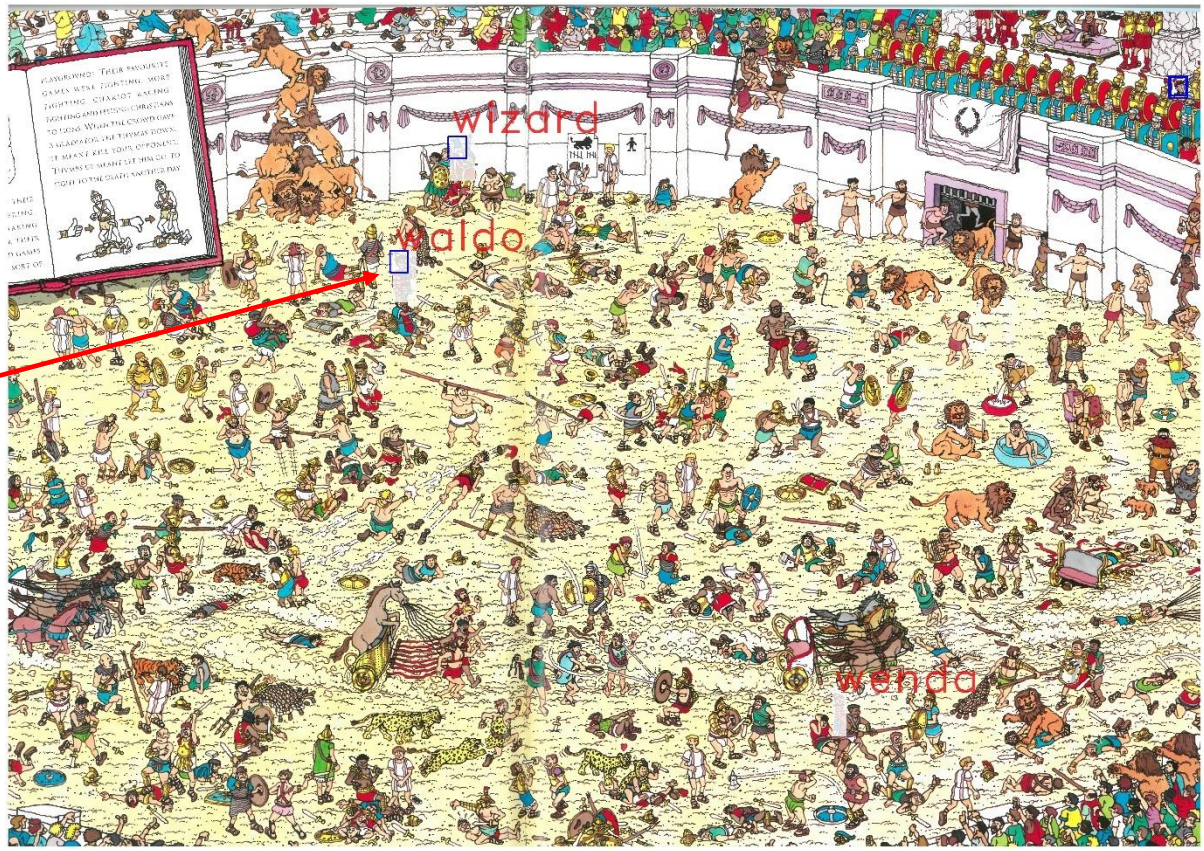


Fig. 14: "011_waldo.jpg" – Image produced for a Waldo detection (Although wizard was detected as well)

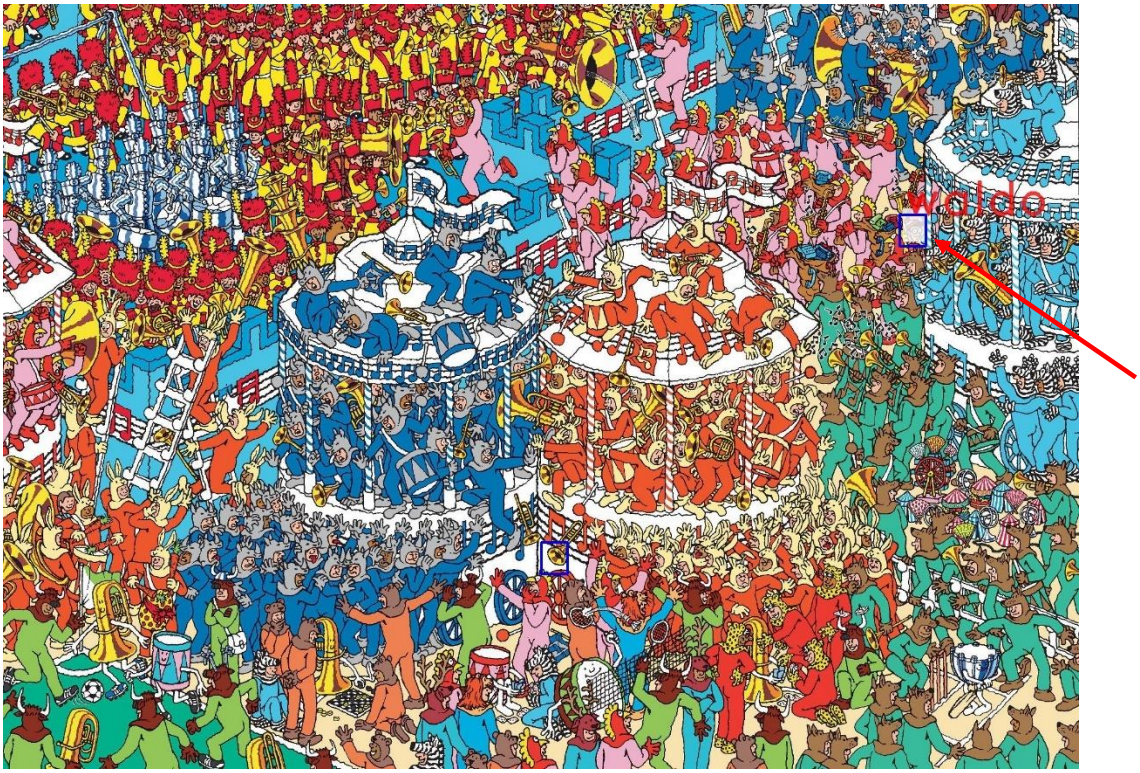


Fig. 15: "039_waldo.jpg" - Image produced for a Waldo detection (Other false positives were detected)

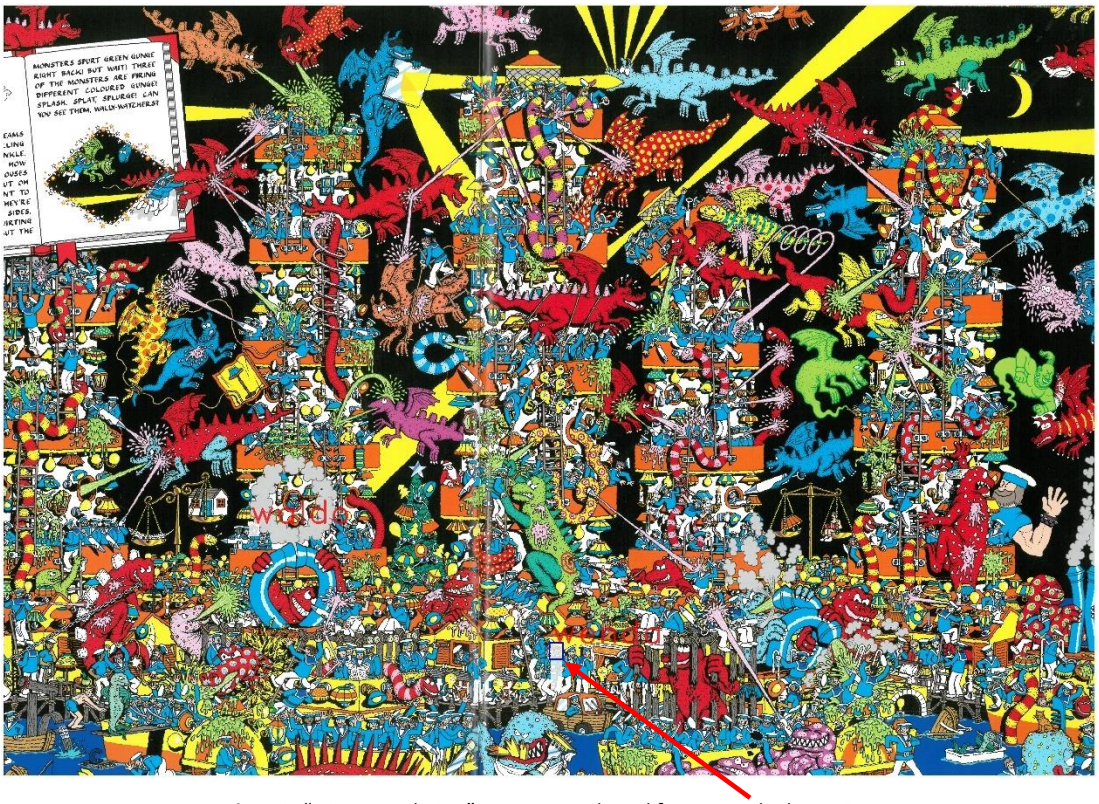


Fig. 16: "031_wenda.jpg" - Image produced for a Wenda detection

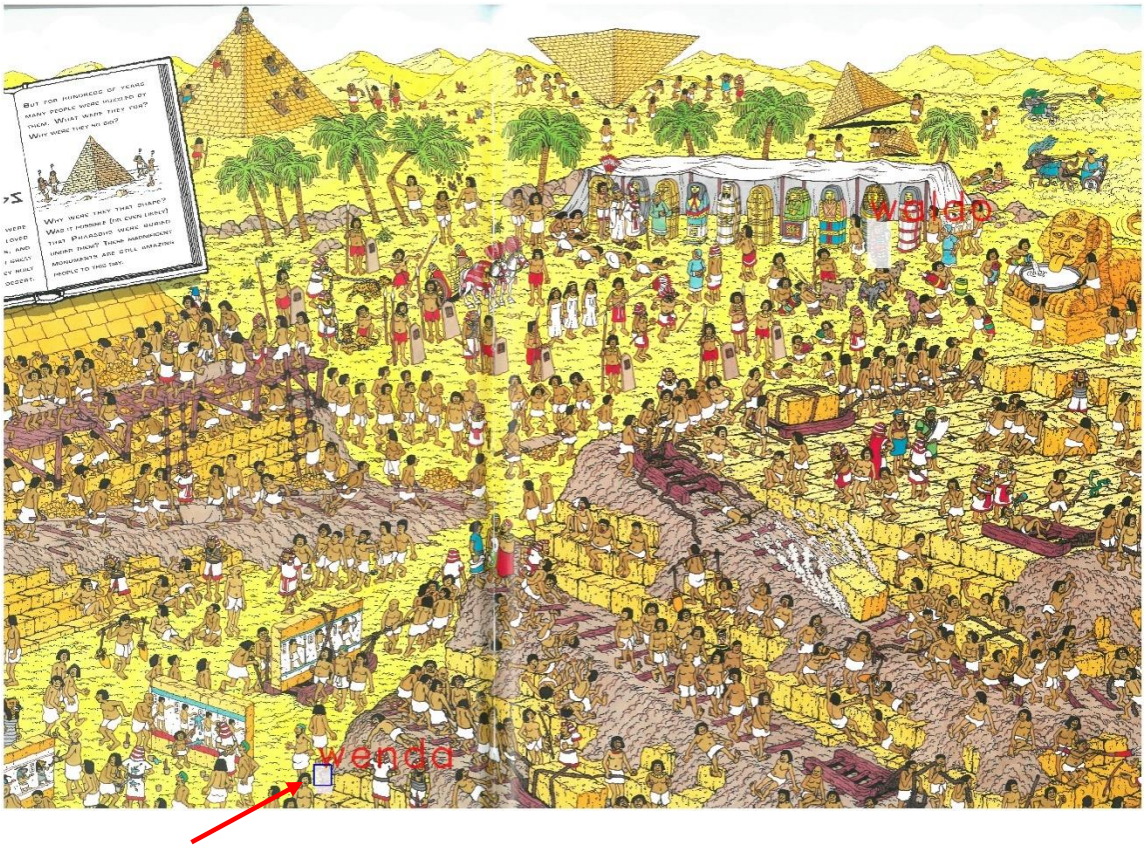


Fig. 17: "014_wenda.jpg" - Image produced for a Wenda detection

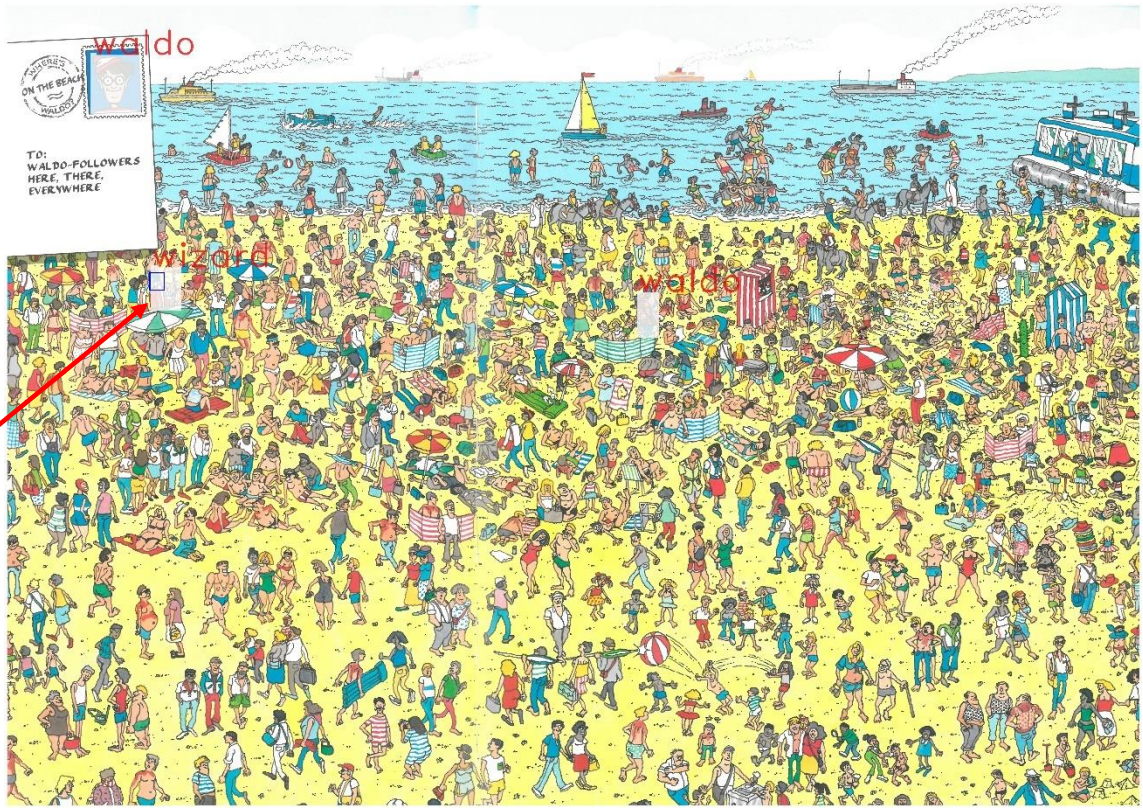


Fig. 18: "044_wizard.jpg" - Image produced for a Wizard detection

Negative detections:

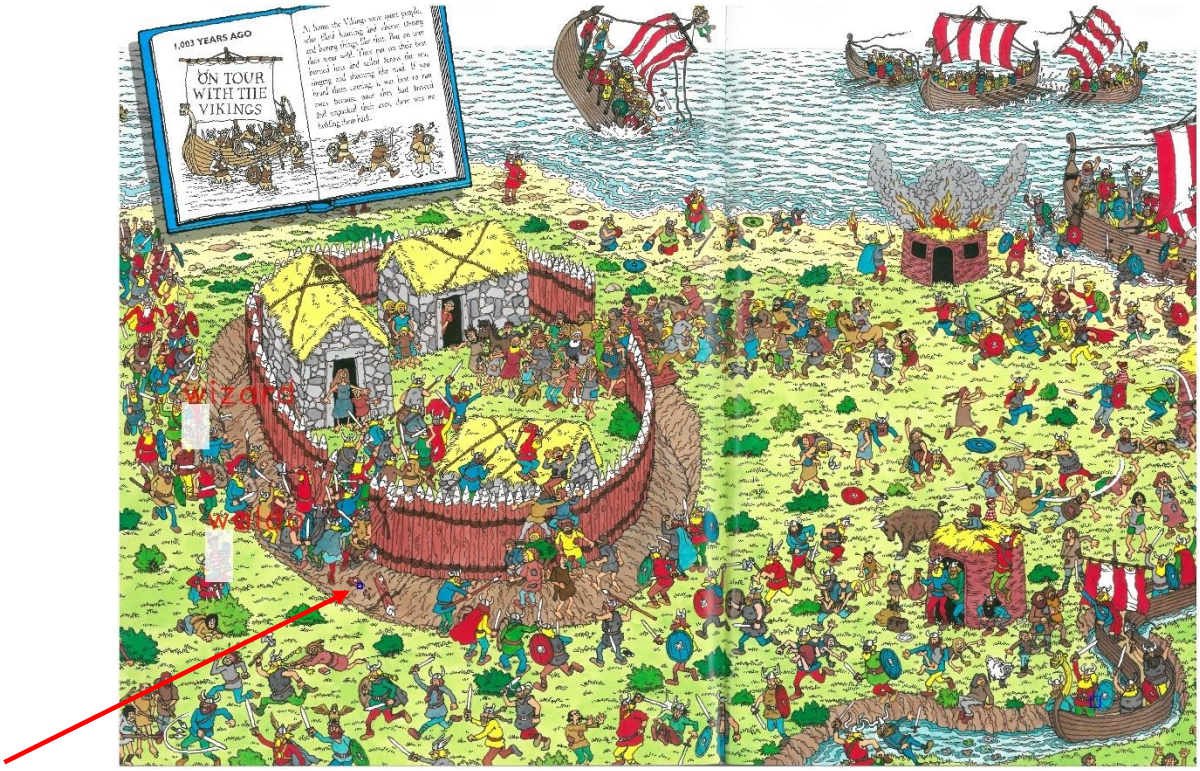


Fig. 19: "008_waldo.jpg" - Image produced for a Waldo detection

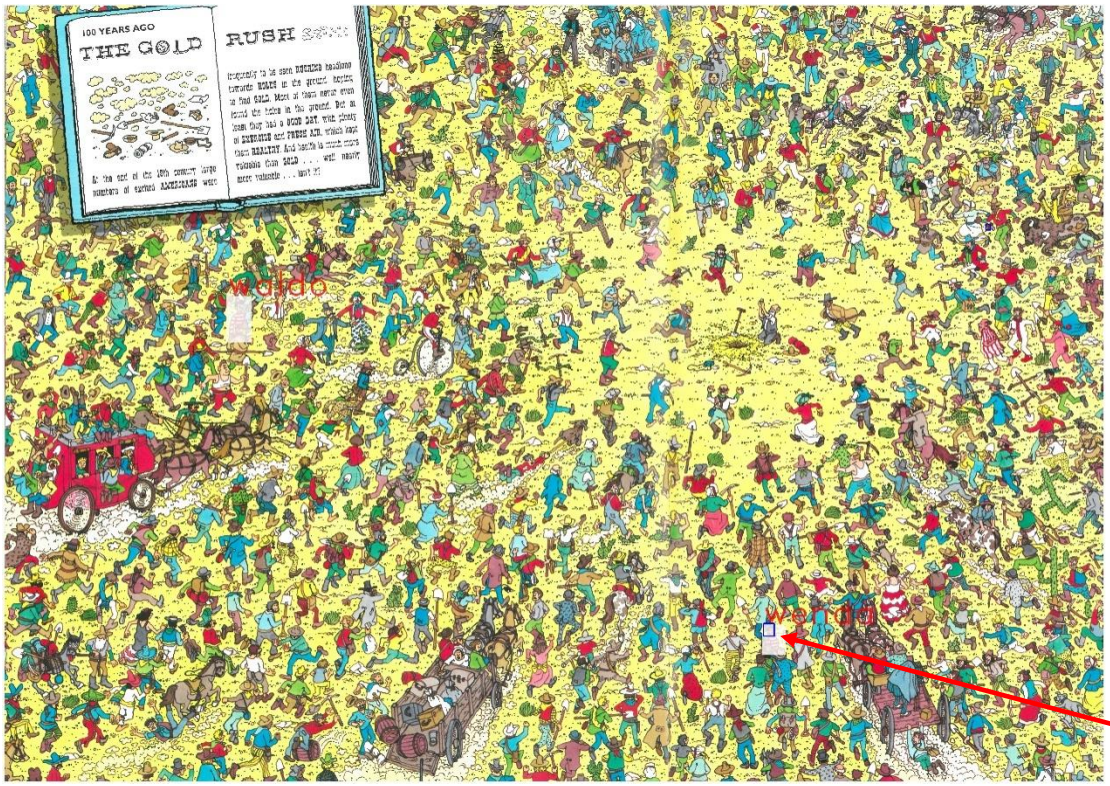


Fig. 20: "063_waldo.jpg" - Image produced for a Waldo detection (But detected Wenda instead)

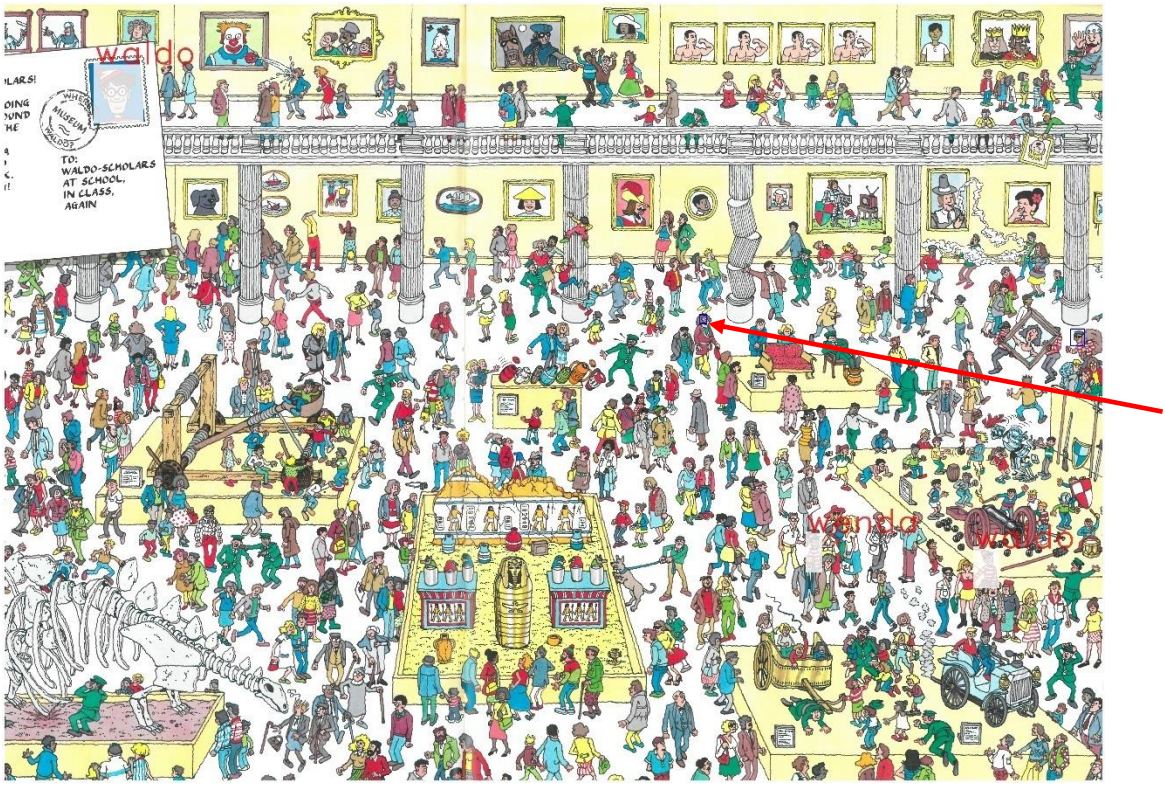


Fig. 21: "049_wenda.jpg" - Image produced for a Wenda detection

Fig. 22: "058_wenda.jpg" - Image produced for a Wenda detection

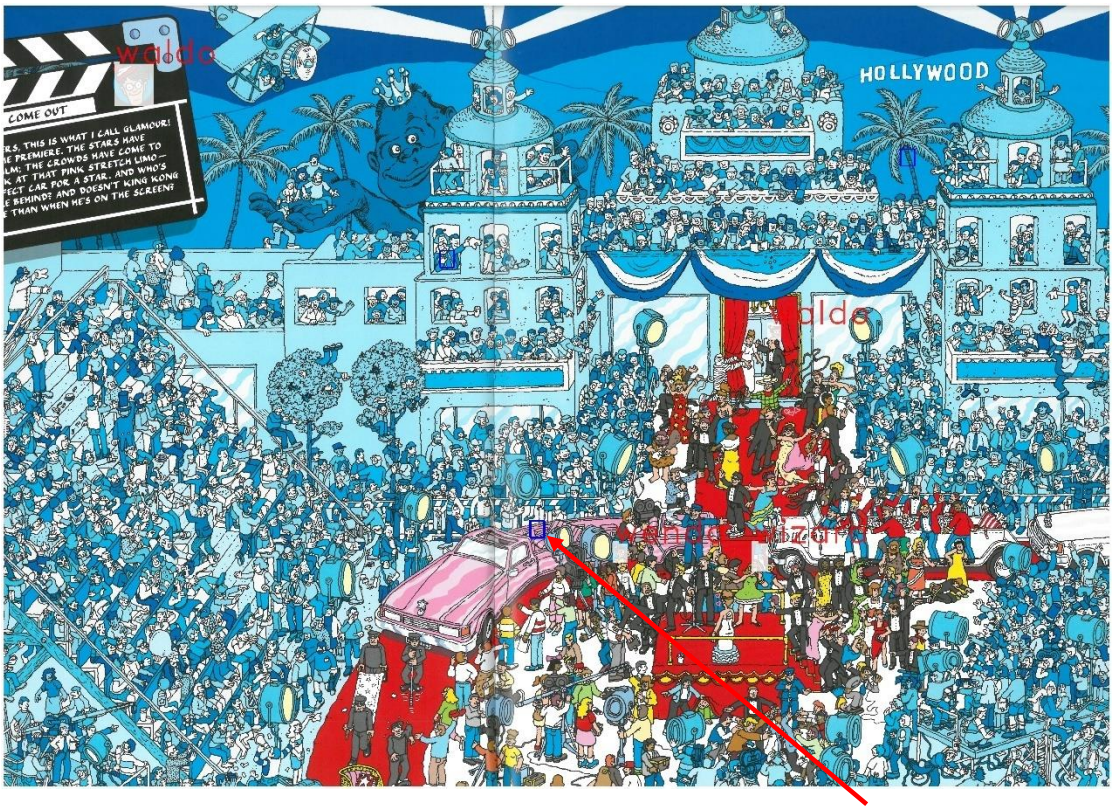


Fig. 23: "079_wizard.jpg" - Image produced for a Wizard detection

The full set of output images from our algorithm execution can be found in the "output_img" folder of the "workspace" folder in the submitted files. File names are all appended with "_<character name>" (i.e. 019_waldo, 019_wenda, 019_wizard), if the file has the character name as the suffix, it means that the detections contained in the image were generated from the SVM detecting that particular character, hence the detections are only supposed to be for that character. If there are no output image files for a particular original dataset image id, it means that the algorithm did not manage to detect that particular character. If there are no output image files for a particular image id, it means that the algorithm did not manage to detect any character from the original dataset image.

Performance Statistics:

Looking at the output text files generated after our algorithm executed over all 70 dataset images, we have come up with the following table:

Character	No. of Images with Character	No. of Images where Character has at least 1 True Positive	Percentage of images where characterhas at least 1 True Positive
Waldo	66	7	10.6%
Wenda	41	17	41.5%
Wizard	25	7	28.0%

Table 3: Algorithm Performance Statistics

We decided not to use the popular Precision and Recall metric to evaluate our algorithm's performance, because we realized there are some very huge imbalance in the datasets, particularly those image scenes where the characters' faces repeat tens or even hundreds of times (i.e. 048.jpg, 068.jpg), hence if only a handful of True Positives were detected in those images, the overall False Negatives over all 70 dataset images will still be pushed to a very high value, and Recall metric may not give an accurate reflection of the strength of the algorithm equally over the whole dataset range.

Furthermore, we admit that we did not have the intention to design the algorithm to detect the very large Character faces in the top postcards or side scrolls of the dataset images, hence we also hope to omit the consideration of the large character face detections in our metric.

Hence we thought that the simple percentage of images where each character has at least 1 True Positive detection would show that our algorithm does make a reasonable amount of True Positive detections over the whole dataset range. The percentage of images for each character where at least 1 True Positive detection is made ranges from about 10% to 42%, as shown in Table 3.

3.3 Discussion

From the range of output images that visualize the detections in the "output_img" folder, our algorithm seems to perform accurate detections better on lower resolution images, as well as images with brighter backgrounds. This could be due to our positive character templates for each character having generally lighter/brighter backgrounds. Also by fixing all templates to be 50 x 60 pixel dimensions and then scaling it up and down during iterations, templates that were cropped from higher resolution images might have lost some fine details that could have helped the HOG and SVM made a better prediction.

Another weakness with respect to higher resolution images such as most of the dataset images (~10000 x 7000) resolution is that there are generally more interest points generated, and when the algorithm starts from a high scale factor (2.5) and gradually scales down, there is higher likelihood that an interest window that doesn't correspond to a true positive may be similar to a positive template, hence it is more difficult narrow down to 1 perfect detection, or even guarantee any true positives.

A strength of our algorithm is that it does make detections that contain true positives in a sizable number of images of varying range of backgrounds, possibly due to the range of negative sample windows obtained from the cropped negative scenes (depicted in Fig. 12). But a corresponding weakness regarding background is that if the background is very similar to iconic features of the character it is detecting, there will be a lot more interest points generated, such as in Fig. 22 above where the background is mostly blue, and the technique to extract interest points for Wizard is to use his blue hat, hence this gives much more interest points, and thus higher probability of a false positive window to be detected as similar to Wizard's face.

3.4 Experimentation with added features

Other experimentation techniques we tried were firstly, to crop away most of the background of the character templates in the Preparation Step, to see if it improves the accuracy of detections of each character, because as mentioned in the discussion, background of the character templates affects the detections in varying range of backgrounds in the dataset images. However, using the cropped templates resulted in a significant deterioration of true positive detections. Hence, we did not include this in our final submitted implementation.

Another technique we tried was to append Discrete Cosine Transform (DCT) features of the character templates and extracted sample and test windows to their respective HOG feature vectors, thus creating a longer vector comprising of both HOG and DCT features for each template/window, hopefully to further differentiate feature vectors from each other, and more accurately train the decision boundary in the SVM. However, implementation with both HOG and DCT features yielded almost no true positives for most of the dataset images that we tried it on. Hence, we also did not include this in our final submitted implementation.

We also tried the popular 'eigenfaces' method to detect the faces of the characters and then appending these 'eigenface features in the feature vector. Eigenfaces involve linearizing the images and then putting them in the same array, then normalizing the images by subtracting the mean of the images from each image and then calculating the covariance matrix by multiplying this matrix with its transpose. Then the eigen-values and eigen-vectors were calculated for this matrix, and those eigen-vectors corresponding to 'n' maximum eigen-values were chosen as the features representing that person. Again, with these added features yielded poor results and we decided to not have them in the feature vector.

Towards the end we also tried to detect the circular glasses of Waldo using an approach called the Circular Hough Transform. If the window had 2 circles detected that would mean there's a higher chance of there being Waldo. This could have been used when we would be left with only a few candidates. However, this method involved tuning of parameters which turned out to be different for each image and hence we had to drop these method because it yielded poor results with some global parameters. Also, sometimes the results were poor as shown below in Fig 24.

Fig. 24: Glasses detection in Waldo

4 Conclusion

In this project, we set out to create a computer vision solution to detect Waldo and his friends Wenda and Wizard. Our detection algorithm comprises of multiple stages of image processing, namely applying color analysis and mask to extract interest points from each dataset image, then extracting negative sample windows from dataset backgrounds, and at interest point locations from the color mask, then generating positive samples of HOG feature vectors from character templates , and negative samples of HOG feature vectors from the sampled negative windows, subsequently training an SVM to binary classify the HOG feature vectors from the interest windows as to whether each contains the character in question or not, and outputs the positive detections in image form (images in “output_img” folder), and text form (text files in “output_txt” folder). This process is repeated at different scales for each dataset image, for all dataset images. At each stage, features are extracted to narrow down the scope of detection, until finally the minimal set of windows are obtained from the SVM, if any, for each dataset image.

The algorithm detailed in this paper is able to achieve reasonable results of **10% to 42%** for thepercentage of images for each character where at least 1 True Positive detection is made, and has the potential of further fine-tuning to improve the detection performance, as there are many parameters that affect each stage, as detailed in Section 3.1.

- END OF REPORT -