UROP 1100 Report

Topic: Augmented Reality Technology for Visually Impaired

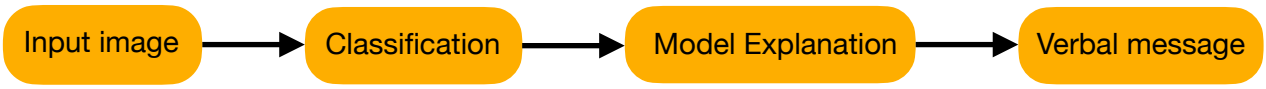
Student: Mu-Ruei, Tseng

Abstract:

Visual impairment is one serious problem that many people faced. Different people may suffer at different degrees and require various levels of medical treatments. Medical treatments, however, are relatively expensive regarding the price and not everyone is able to afford the cost. As technologies become more and more advanced in modern society, it opens up new approaches to conquer this problem. The method that we proposed is to apply augmented reality techniques on wearable devices so that we can largely reduce the costs and also risks for those that are needed. This report will discuss the methodology of implementing this idea.

Introduction:

People have been using deep neural networks for image classification for quite a long time. By passing images through some convolution and fully connected layers, we are able to determine the item in the picture. The first major task for this project is to capture images and identify all the items inside with a high accuracy. However, if we directly give the classification output to visually impaired people, they might show distrust in our result. It is not enough to just provide what things are in the picture given the accuracy. We have to come up with evidence to support our classification so that they can better accept the output. In order to convince visually impaired people, we will construct a saliency map for the image and also determine the shape of them. After we obtain these informations, we can finally produce verbal information and deliver the result. Here is the work flow for the project:



Implementation:

1. Dataset:

The dataset we are using is ImageNet. ImageNet is a large visual database designed for use in visual object recognition software research. It provides more than 1.2 millions images with 1000 numbers of categories for training data. ImageNet database is organized according to the WordNet hierarchy where each node of the hierarchy is depicted by hundreds and thousands of images.

Fig 1

2. Classification

For the image classification network, I am using Inception-v3 backbone and using pre-trained weights for transfer learning. Inception network is a convolutional neural network that has fewer parameters compared to other networks like AlexNet and VGGNet. It consists of 42 layers with lower error rate that makes it the 1st runner up for image classification in ImageNet Large Scale Visual Recognition Competition. Inception v3 unlike previous versions introduced the concept of factoring convolution in order to reduce the network parameters without decreasing the network efficiency. For example, one 5x5 convolution can be replaced by two 3x3 convolution layers since they have the same receptive field. 5x5 convolution will have 25 parameters while two 3x3 convolution only requires (3x3)x2 = 18 parameters. Besides factoring into symmetric convolution, it also proposes optimizing methods by factoring into asymmetric convolution. To illustrate, one 3x3 convolution layer can be replaced by using one 1x3 followed by one 3x1 layer. This can reduce the number of parameters from 9 to 3+3 = 6. Less number of parameters reduces the possibility for the network to overfit so the network can be further extended in depth.

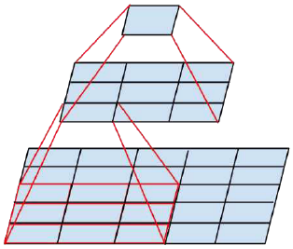


Fig 2.1

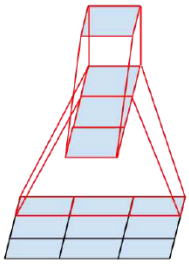


Fig 2.2

Inception-v3 network also adds one auxiliary classifier on top of the last 17×17 layer, where accuracies are nearing saturation. The classifier includes batch normalization and dropout operations to serve as a regularizer to the network. Inception-v3 requires input images of size 299x299; therefore, before passing our image to the network, we have to do some preprocessing on the images. I make use of transform from torchvision library to resize the images to be in uniform size of 299x299, then normalize the color channels for every image to prevent gradient explosion.

Here is the prediction output for the Incpetion-v3 network.

Fig 2.3

3. Model Explanation

3.1 Explainable AI

After obtaining correct image predictions using the Inception-v3 network, the next goal is to capture evidence to interpret our classification output. The technique we tried to use is explainable AI. People usually view Deep Neural Networks as a black-box model. When facing decisions that affect human's lives, there is a need for understanding how decisions are made by the machine. Explainable AI focuses on machine learning techniques that allow users to understand and trust the model output. Common approaches to explain Convolutional Neural Networks can be divided into two following categories:

1) Try mapping back the output to the input space to show parts that are discriminative for the output
2) Investigate inside the network and interpret how the intermediate (hidden) layers see the external world

Here we study more in the techniques from the first category, to be more specific, class activation mapping (CAM) and saliency map. A class activation map for a particular category indicates the discriminative image regions used by CNN to identify that category. It is implemented by substituting the last fully connected layers to a Global Average Pooling (GAP) layer. The GAP layer will average the activation map of each feature and concatenate them as a vector output. Weighted sum of this vector will then be calculated and the result will be fed to the last softmax loss layer. The important regions of the image can be shown through back projecting the weights of the output on the feature maps.

Saliency map, on the other hand, specifies parts in the input image that contributes the most to a specific layer. It is one of the most frequently used methods for generating interpretation and also the approach that I used in this project. A saliency map for a specific image can be generated by performing a backpropagation algorithm and compute the gradient of the logits with respect to the input of the network. Using backpropagation, we can highlight pixels of the input image based on the amount of the gradient they received.

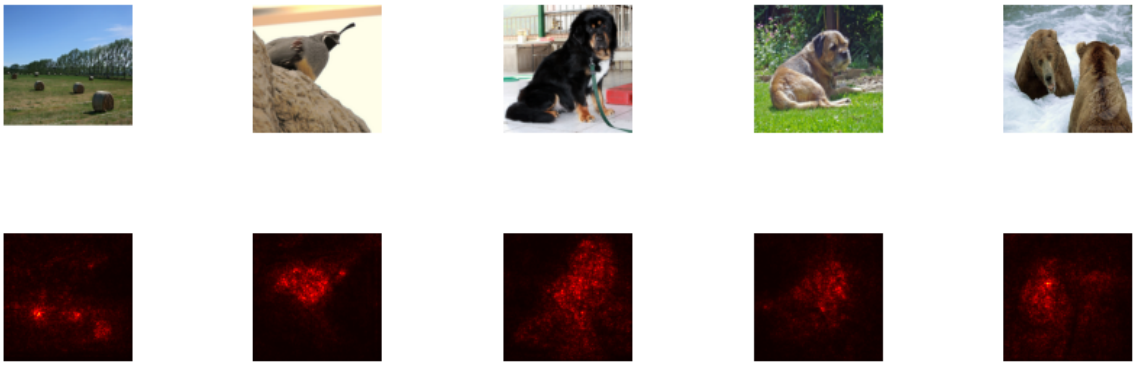


Fig 3.1

As Fig 3.1, we can see that the saliency map identifies areas that contributed to the image classification result. However, it is relatively difficult to describe the distribution of high intensity points to those visually impaired people. In order to help them better understand, we can determine the shape of those high intensity points in the saliency map. For example, for the image of a quail, we have to show that high intensity points form the shape of a triangle.

3.2 Shape detection

The main idea we used to estimate the shape of the saliency map is to apply findContours method in the OpenCV library and approximate its polygonal curve. The shape is determined by the number of approximate curves that are generated. If the selecting contour has three approximate curves, this implies that the area might be in the shape of a triangle. One major difficulty for this method is that it requires contours to be as continuous as possible but the saliency map we generated are discrete points. We cannot directly feed the output to the shape detector. We tried several approaches to conquer this problem. We first perform convolution on the saliency map and apply global thresholding. We tried to only determine the shape for those areas that have relatively high intensity. Global thresholding helps reduce noises in the map. We had tried using different thresholds and kernel sizes.

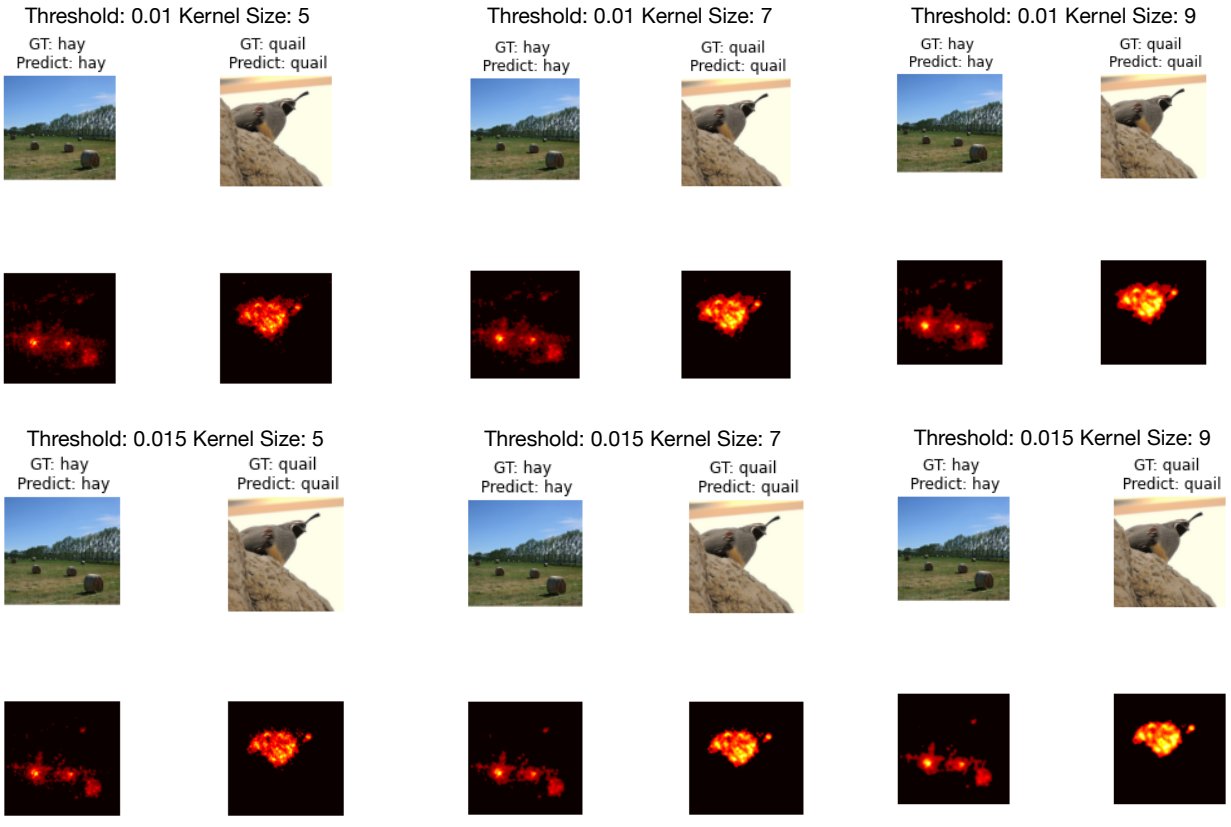
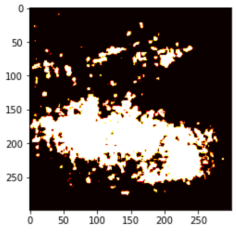


Fig 3.2.1

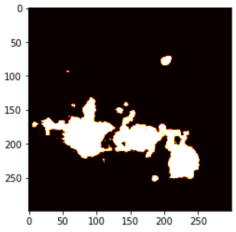
As we can see from Fig 3.2.1, when increasing the size of the kernel and increasing the threshold value, points become more concentrated. Here is the shape estimation for the hay image using threshold 0.01 with kernel size 5 and threshold 0.015 with kernel size 9. A better thresholding method largely reduces the number of shapes detected (Fig 3.2.2).

Threshold: 0.01 Kernel Size: 5



circle circle square circle circle circle
circle circle circle circle circle pentagon
circle circle circle circle circle triangle
pentagon circle circle pentagon circle circle
pentagon circle circle circle pentagon circle
circle circle circle circle circle pentagon
circle circle rectangle circle circle pentagon
circle triangle circle circle triangle circle
circle circle circle pentagon triangle pentagon
circle circle circle rectangle rectangle circle
triangle circle pentagon pentagon circle circle
circle pentagon circle circle circle circle
rectangle circle rectangle circle circle rectangle
pentagon circle pentagon circle circle pentagon
rectangle circle circle pentagon pentagon circle
circle rectangle circle circle circle pentagon
circle circle circle pentagon circle circle
circle circle circle circle pentagon rectangle
pentagon circle pentagon circle triangle

Threshold: 0.015 Kernel Size: 9



pentagon circle pentagon circle circle circle
circle circle circle pentagon circle triangle
circle circle circle pentagon pentagon circle
circle

Fig 3.2.2

Even though tuning the threshold for the saliency map smoothen the contour and reduces additional shapes detected, it is still not enough. Also one problem that the global thresholding method will face is that a single threshold cannot apply for all images. Different images may have different intensity in its saliency map, leading to situations where thresholding improves performance in one image but worsens the other. In order for thresholding to better apply to all images, we have to lower the thresholding value, having the side effect of getting more points and noises.

We also try to apply clustering on the saliency map to see if we can separate the high intensity points into different clusters and only choose those with large areas. The clustering method we used is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSAN is a density-based clustering non-parametric algorithm that groups together points with many nearby neighbors. It will mark points having few neighbors as outliers. The clustering result can be determined by the distance (ϵ) and minPts. Fig 3.2.3 is one clustering comparison between different distance (ϵ). For

the hay image, we are able to separate one hay from others using a smaller distance, but we still cannot separate the other two.

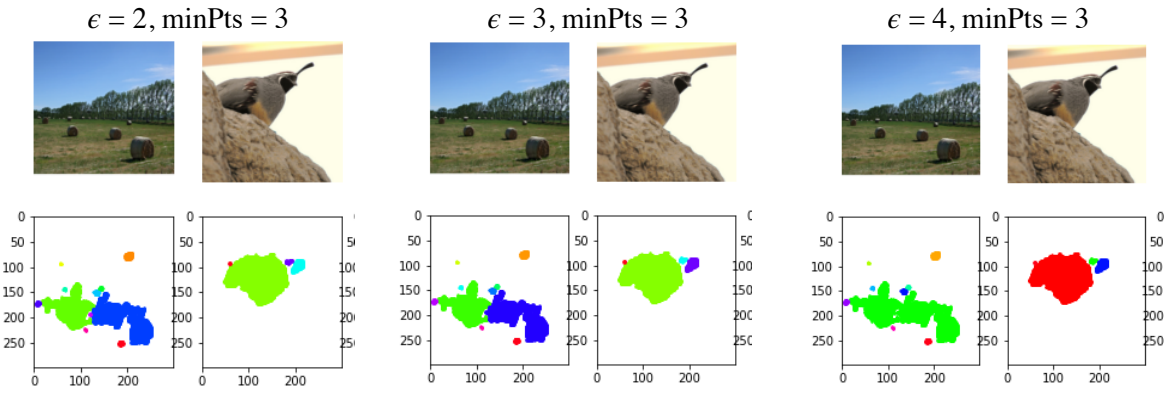


Fig 3.2.3

Since the shape detection result is still not satisfying by using the previous results, we try to use image segmentation on the original image instead. We select segments that contain high intensity points in the thresholded saliency map and use the contour of the segment to estimate the shape. This method different from previous experiments is that the contour generated from image segmentation can be more continuous. However, the disadvantage of this method is that it requires longer time to calculate image segmentation. The segmentation method we currently use is resnet101 with only 21 numbers of classes. We will have to train the segmentation network with 1000 number of classes to cover all the classes in ImageNet. Here is the segmentation result for the current network. The contour line is continuous and the shape detector is yet under experiments.

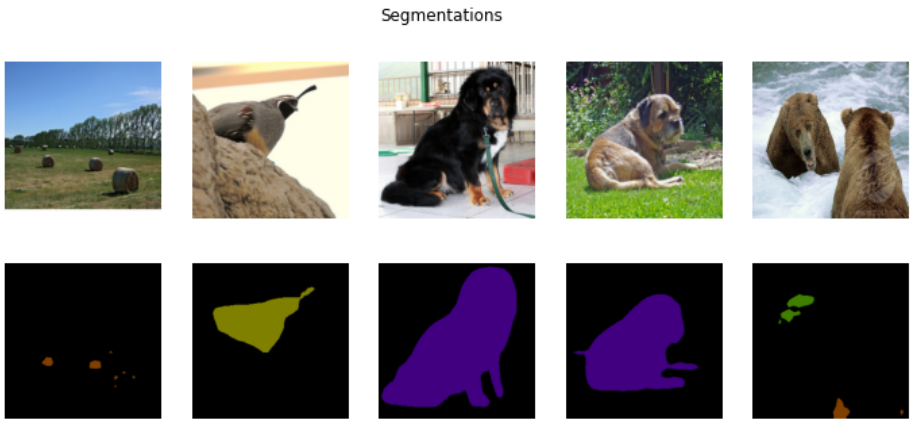


Fig 3.2.4

4. Verbal Message

Since we are currently dealing with the problem of estimating shape property, the generation of verbal message has not been covered.

Future work:

The future work for this project is to continue finding methods to better estimate the shape for the high intensity points in the saliency map. The segmentation method can be further improved to 1000 number of classes so that it can perform segmentation on ImageNet dataset more accurately. Also, considering the run time, we might want to improve the first thresholding method to be more robust. We can try to use different threshold values for different images based on its distribution in the intensity range. After obtaining the shape information, we can collect other data like colors, size, depth of the object in the image and generate verbal messages to the visually impaired people so that we can tell them what is in front of them and convince them with the idea.

Summary:

This project is dedicated to building a system that not only classifies images for those visually impaired people, but also shows the evidence and makes the model more transparent so that we are able to convince them to trust our prediction result. Through this project, I become more familiar with data preprocessing and setting up image classification models. This is also the first time that I was introduced with the idea of explainable AI. I originally thought of deep learning neural networks as a black box, not until now that I am able to visualize and interpret the activation map to figure out pixels that contribute to the classification output. Shape determination is also an important part of this project. I have to come up with different solutions to solve the problem. This further strengthened my knowledge in the realm of clustering and also image segmentation. Although the current shape detection isn't accurate enough, I believe that if having more time investigating, the result can become more correct. Overall this is a heavy workload but a fruitful project, and I have learned a lot during the working process.

References:

[1] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J, Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. Computer Vision and Pattern Recognition. Retrieved from https://arxiv.org/pdf/1512.00567.pdf

[2] Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot. A, Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F. (2019) Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. Computer Vision and Pattern Recognition. Retrieved from https://arxiv.org/pdf/1910.10045.pdf

[3] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A. (2015) Learning Deep Features for Discriminative Localization. Computer Vision and Pattern Recognition. Retrieved from https://arxiv.org/pdf/1512.04150.pdf