Loki

Index:

Problem Statement:

Around 64% of the population regularly wears eyeglasses. Out of this group, approximately 14.1% of those who purchased eyeglasses in 2020 opted to buy them directly online. Moreover, about 30% of these 2020 eyeglass buyers utilized the internet to compare prices, explore various eyeglass styles, or identify the eye care professional (ECP) or retailer from whom they intended to make an in-person purchase later.

Given the prevalent use of the internet in today's context, it has become increasingly crucial to offer customers the option to purchase eyeglasses online that can be accurately assessed and tested. While the market has successfully addressed the former aspect, it has faced significant challenges in delivering precise IPD (Interpupillary Distance) measurements. To address this gap, there is a pressing need for a model capable of determining a person's IPD distance using image or video analysis. Such a model would provide users with their IPD measurement in millimeters, enhancing their online eyeglass shopping experience.

Solution:

Using AI and ML, our app is going to find out the IPD distance of a patient. The patient opens the app, places his/her credit card on their forehead and clicks a picture. After the picture is taken, The model analyzes the image and gives the IPD value as the final output.

Introduction:

What is IPD?

IPD stands for Interpupillary Distance, and it refers to the distance between the centers of the pupils in the two eyes. It is an essential measurement in optometry and ophthalmology as it helps determine the optimal fit for eyeglasses, contact lenses, and other vision-correcting devices.The IPD distance can vary from person to person and is typically measured in millimeters (mm). The **average adult IPD** is around **58-68 mm**, while for children, it can be slightly smaller.

Why do we need IPD?

IPD is crucial for achieving proper binocular vision. It ensures that both eyes can focus on the same point and work together efficiently. When using **eyeglasses** or certain optical devices, proper alignment of the pupils with the optical centers of the lenses is essential to provide the best visual clarity and comfort. Apart from daily use, IPD is used for viewing 3D images or movies, i.e It is used to adjust the convergence of the images to match the viewer's eye position.In VR and AR applications, knowing the user's IPD is crucial to calibrate the display and provide an immersive experience.

How do we measure IPD?

There are few traditional methods available in the market that help us to find the IPD of a patient's/customer's eyes. Few of them are

Pupilometer: An instrument called a pupilometer or pupillometer is often used to measure the IPD accurately. It uses infrared light to detect the pupils' positions and calculates the distance between them.

Mirror Method: In a mirror method, a person looks at their own pupils' reflections in a mirror. A ruler or measuring tape is then used to measure the distance between the reflections.
Apart from these, optician/optometrists find IPD using **comprehensive eye assessment**.

Why bring Technology?
Though these methods give accurate results, growing technology needs fast and accessible testing. That is the reason we are coming up with a technology where the users can test their frames and order specs online. So for this, users have to go to the frame checker website of LVPEI and put their credit card on their forehead. The application takes in the photo and determines the IPD distance which next leads to checking frames accordingly.

Working Principle of the AI model:

We all know that the length and width of credit cards are the same throughout the world. Generally, if we use image processing and find out the pupil distances, we get the value in terms of the number of pixels. To make the decision simpler, we use a reference object whose length is already known. The reference object's length is calculated in terms of the number of pixels and pixel per mm is found out.

Pixel per mm(ppmm) = Length in mm / Length measured in terms of pixel

After getting the pixel per mm ratio, the ppmm is multiplied to the final IPD distance that we get in terms of number of pixels and obtain the IPD in mm

IPD (in mm) = IPD(no of pixels)* ppmm

This project was first given to a consultant. From them we have got the code for measuring IPD but the actual IPD and measured IPD differ by 1-2 mm which is too crucial and can't be neglected. One of the major concerns that we had was that the code was not able to detect the credit card accurately which is considered a major problem to solve.

My Journey:

I was initially introduced to the project by Mr. Santosh, who is leading the project on credit card detection. My initial focus was on exploring edge detection techniques, such as Canny edge detection, monocular depth estimation, and utilizing shape predictors to track pupils (e.g., Haar Cascades, Hough transform, VGG, etc.). However, my attempts to use these algorithms on downloaded internet images resulted in unsatisfactory credit card detection outcomes.

Upon discussing these challenges with Mr. Santosh, he provided valuable insights for a new approach. His suggestions included examining existing websites that perform Interpupillary Distance (IPD) measurement and reframing the problem from an IPD detection perspective rather than just edge detection.

Incorporating his suggestions, I delved into existing algorithms and websites that focus on IPD measurement. I encountered various code implementations for calculating IPD and ppmm (pixels per millimeter) measurements. To address coding discrepancies, I sought guidance from colleagues like Manogna, who helped clarify coding nuances and differences that I encountered. Additionally, I reached out to an ML enthusiast who had extensive experience with a similar project, and their insights proved invaluable in resolving specific queries.

Applying the collective wisdom gained from these interactions, I continued refining my code. During this process, I realized the critical role of creating a dedicated dataset for training my model. Upon discussing this realization with Mr. Santosh, it became evident that I needed to generate my own dataset.

I proceeded to take consent from interns and captured images to form the foundation of my dataset. Recognizing the necessity of annotated data for model training, I obtained the required permissions and initiated the data annotation process.

Code:

This is the github repo that I created for this project.
Go to src/components and the actual code lies there. I will go through each python file separately.

1. detect_measure_cc.py
   This is the first file that detects the credit card and measures the length of the detected credit card. As I mentioned, I had no dataset in particular so I didn't get the chance to run it. I have used the already existing tensorflow's resnet50 model to train my data. If the dataset is available then annotate the dataset and train the model. The model is available in the '`models/research/object_detection/configs/tf2`' directory of the TensorFlow Object Detection API repository.

   Copy the model and change the following:
   Open the copied configuration file and update the paths and dataset details:
   num_classes: Set this to the number of classes in your dataset, including the class for the credit card.
   fine_tune_checkpoint: Set the path to the checkpoint of the pre-trained base model. This should be a .ckpt file.
   label_map_path: Set the path to your label map .pbtxt file that defines the class mapping.
   input_path and **label_map_path** under **train_input_reader** and **eval_input_reader**: Set the paths to your TFRecord files and label map for training and evaluation.

   You can also apply fine tuning
   Fine-tuning requires setting various hyperparameters and training settings. Key parameters include:
   batch_size: Set the batch size for training.
   num_steps: Set the number of training steps.

initial_learning_rate: Set the initial learning rate for the optimizer.
momentum_optimizer_value (if applicable): Set the momentum for optimizers like SGD.
fine_tune_checkpoint_version: Specify whether to use the V1 or V2 format for the checkpoint.

2. ppmm_cc.py
   The results of the previous file goes to the ppmm_cc.py file and the pixel per millimeter is found out.

3. ipd.py
   The eyes are detected using the haar-cascades algorithm. This inturn detects the pupils of both the eyes. The center of pupils are calculated and distance between them is taken out in terms of the number of pixels. Then, this value is multiplied with ppmm value to obtain IPD in terms of millimeters.

Conclusion:

The primary challenge from the beginning was to accurately detect credit cards, which can be achieved using established image processing models. I opted to employ the ResNet-50 model for this task. However, the project's success hinges on the quality and quantity of the dataset obtained. With a sufficient number of appropriate images in the dataset, training the model becomes a more manageable endeavor.

Future Opportunities:

Another aspect we can focus on is enhancing the accuracy of the Interpupillary Distance (IPD) measurement. Recent studies have highlighted a minor variance of around 2-3mm from the actual IPD value. Even this slight discrepancy can have significant implications for eyeglasses prescriptions and potentially impact the wearer's visual experience negatively.