Extraction of Real-World Objects in the Environment through Computer Vision

Prof. Kavita Jain, Varun Singh, Rohan Talele, Kartik Kashyap

Department of Computer Engineering, Xavier Institute of Engineering, Mumbai University, Mumbai, India

kavita.j@xavier.ac.in
20170155.varunssa@student.xavier.ac.in
20170156.Rohantns@student.xavier.ac.in
20170162.kartikkyr@student.xavier.ac.in

***Abstract* – This paper is designed to shorten the process of transferring images from our phones to our computers through Emailing, Bluetooth, WhatsApp etc. by bypassing all of this. Instead, we would capture the image from our phones, extract only the subject of the image and then transfer that extracted image directly to our computer screen. For now, we will be only considering Adobe Photoshop, an image and graphics editing software as the screen on which the extracted image would be displayed. From there the image can then be further edited in photoshop, copied or saved on the computer. Optical Character Recognition (OCR) and QR code scanning is provided for additional benefit. This type of application would be helpful for editing images by photography professionals or just for transferring images quickly since it would cut down the work required.**

Keywords — Computer Vision (CV), Boundary-Aware Salient Object Detection (BASNet), Boundary Detection, React Native.

INTRODUCTION

SnapCut is a Computer Vision and Machine Learning prototype application that allows cutting elements from your surroundings and pasting them in an image editing software. For this we propose a technique to transfer an image taken from the real-world, removing its background detail and just keeping the image. Then transferring that image of the object to a computer screen by positioning the mobile camera in the direction of the computer screen to paste it.

Objectives:

a. The app is being created using React Native and the expo framework [8]. We plan on keeping the GUI relatively simple since the main focus is on the transfer feature of the app.

Figure 1. (a) and (b) show input and output respectively by the BASNet algorithm

b. To create a technique to transfer an image taken from the real-world, removing its background detail and just keeping the image. We will achieve this using BASNet (Boundary-Aware Salient Object Detection) [2] [3].
c. Then for the second part transferring that image of the object to a computer screen by holding the mobile camera and positioning it in the direction of the computer screen to paste it.

RELATED WORK

Saliency detection aims at modeling human visual attention mechanism to detect distinct regions or objects, on which people likely focus their eyes in visual scenes. Contextual information plays an essential role in this visual task. As one of the earliest pioneering computational saliency models, calculate the feature difference between each pixel and its

surrounding regions as the contrast to infer saliency. Numerous methods have been subsequently developed that utilize local or global contexts as the reference to evaluate the contrast of each image location (i.e., local or global contrast). These models aggregate visual information at all the locations of the referred context region into a contextual feature to infer contrast. First, we searched on boundary detection and saliency detection of images.

BASNet: Boundary-Aware Salient Object Detection

Most of the deep learning methods focus on the region prediction when it comes to saliency prediction but now, they have created a new loss function where the BOUNDARY of the object is also considered [3] [4]. This technique is known as BASNet (Boundary-Aware Salient Object Detection). The network structure of the entire BASNet algorithm, and it is also the pipeline of the entire algorithm is shown in [4].

PROPOSED DESIGN

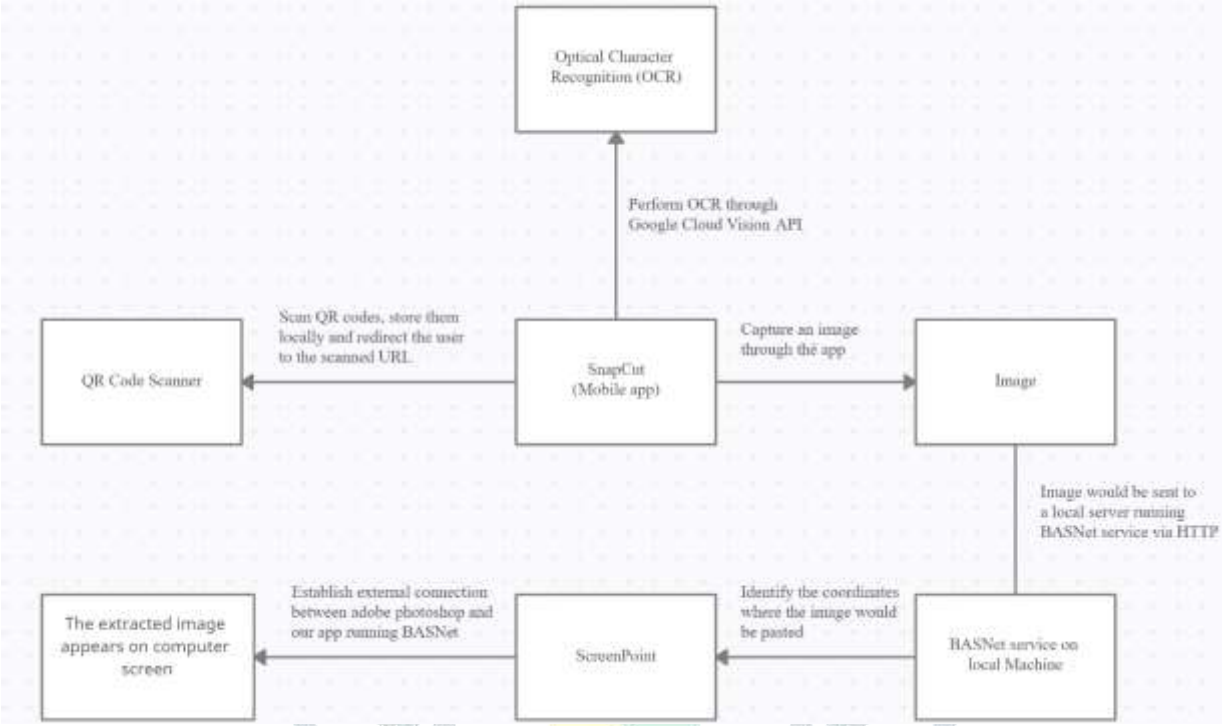

Figure 2. Block diagram of system

IMPLEMENTATION METHODOLOGY

To start from the app design, we had decided early on to make sure our app is cross platform compatible, therefore we had to choose between different stacks namely Flutter and React Native. We ultimately decided to go with React Native since it provided better support for Machine Learning algorithms and required JavaScript for coding which we were familiar with.

This Prototype will have 3 independent modules as follows:

The Mobile App

The Base Application will be developed using React Native for multiplatform support. It will provide the user with the ability to extract objects from the environment and transfer them to their computer screens, perform OCR and scan QR codes.

The Local Server

This will be interface between mobile app and Photoshop. (We will be using Adobe Photoshop Software for Pasting Purpose). For finding the Position Pointed On screen by the Cameras we have used Screen Point, an OpenCV SIFT library for detecting the centroid of a screen. BASNet enables the app to detect the object and remove the background, while SIFT matches coordinates on the phone with the computer screen allowing you to place digital captures in specific

positions across your computer screen. This will Find the (x,y) co-ordinate of the Centroid of an image Pointing at another Image.

Object Detection/Background Removal Service

Now we will be Using BASNet. It performs Object Detection and also separated the object image from the Background image. Now for using BASNet we have to deploy model on local server as an Http service. All the Images will be Pasted in Adobe Photoshop.

We need to run and execute the HTTP service wrapper for BASNet on our machine. Ideally the BASNet service should be running on a dedicated server but for this paper we are enabling our own PC to be the local server as well as the client.

Figure 3: (a), (b) and (c) all show different outputs of the extracted image.

RESULTS

The SnapCut app created in React Native has the basic requirements of an app such as the ability to login, navigate, share and so on. Below are some screenshots of how the app looks.

Dashboard

Figure 4: Shows the dashboard

The dashboard screen is like the main menu of the app with further navigation options to other features and areas of the app. It shows the name of the user who has logged in currently and has the option to logout. From the dashboard the user can select the extract and paste option, perform OCR and scan QR codes. It also has a share icon for sharing the app and a help icon for instructions on how to setup the cut paste feature on the user's computer.

OCR (Optical Character Recognition)

Figure 5: Optical Character Recognition screen

The OCR screen performs extraction of text from images with the options to select an image from gallery or capture an image and perform OCR on it. It also shows a preview of the image selected previously. The user can also copy the extracted text to their clipboard or directly share it through email, WhatsApp and more.

Login and Signup

Login using email, your mobile phone through OTP is provided. Along with this biometric login is also provided with fingerprint login, your mobile phone's face recognition system or the pattern/pin that you have set up. Other basic requirements such as resetting of password if the user has forgotten it is also provided.

Paste to Computer

The Paste to Computer screen does the extraction and transfer of objects to the computer screen. The setup of this requires the user to have photoshop on their computer and rest they need to follow the instructions in the help screen for setup. This feature requires the IP address of the user's computer to establish connection between the BASNet service and the server running on the user's computer and the mobile app.

QR Code Scanner

Figure 6: Scanned QR Code details

The Scanner screen has the ability to scan QR codes provided it is within the frame, the icon present at bottom is used to direct the user to the history page which stores the records of the QR codes scanned previously. From here the user can view the details of the QR code scanned with options to copy the URL link or redirect the user to the link itself or delete the record altogether.

FUTURE SCOPE

This prototype app SnapCut has been built with the ability that allows cutting elements from your surroundings and pasting them in an image editing software, along with this we have also incorporated a QR Code Scanner and OCR functionality in our app. The extraction process could also be moved towards the cloud to ease our processing using software such as Runaway ML. The ability to store extracted images, texts and scanned QR codes to the cloud can also be added. Of course, even more entirely new features can add to enrich the apps functionality even more.

.

CONCLUSIONS

We have achieved what we set out to do by developing this paper which is designed to shorten the process of transferring images from our phones to our computers through Emailing, Bluetooth, WhatsApp etc. by bypassing all of this. Instead, we capture the image from our phones, extract only the subject of the image and then transfer that extracted image directly to our computer screen. For now, we are only considering Adobe Photoshop, an image and graphics editing software as the screen on which the extracted image would be displayed. From there the image can then be further edited in photoshop, copied or saved on the computer. Texts can be extracted from images and QR code scanning is provided for additional benefit. This type of application has proven to be helpful for editing images by photography professionals or just for transferring images quickly since it has cut down the work required.

REFERENCES

[1] https://github.com/NathanUA/BASNet

[2] https://openaccess.thecvf.com/content_CVPR_2019/papers/Qin_BASNet_Boundary-Aware_Salient_Object_Detection_CVPR_2019_paper.pdf

[3] https://medium.com/@SeoJaeDuk/cvpr-basnet-boundary-aware-salient-object-detection-a025c7f7d94f

[4] https://openaccess.thecvf.com/content_cvpr_2018/CameraReady/1251.pdf

[5] http://tutorialspots.com/centos-7-how-to-install-basnet-http-server-6004.html

[6] https://www.visityork.org/explore/national-centre-for-birds-of-prey-p795431

[7] https://drive.google.com/file/d/1s52ek_4YTDRt_EOkx1FS53u-vJa0c4nu/view

[8] https://reactnative.dev/