SKETCH2CODE

Muhammad Ebadullah Khan, Muhammad Huzaifa Abid, Farah Yaqoob, Muhammad Zain
Department of Computer & Information System Engineering
NED University of Engineering & Technology

Abstract— The first step in creating an application is to sketch on paper a wireframe that describes the structure of the user interface. Designers face challenges in converting wireframes to code. This often requires handing off the design to a developer, who then implements the graphical user interface (GUI) code. This work is time consuming for the developer and therefore expensive. In this project the authors solved this problem. Built an application that converts sketches directly into code. This project uses computer vision techniques to solve this problem and train a model to generate this UI interface code. For training, datasets were obtained manually, and the authors used data augmentation on the datasets. This project uses both the Tensorflow object detection algorithm and the Yolo algorithm to detect objects in sketches, such as buttons and text, and translate them into a programming language. This project also uses other computer vision techniques to recognize colors and handwritten text in user interface sketches. Users simply take a picture of their sketch and upload it to the application. The algorithm works on the backend and makes the code for this sketch available for users to download. The goal of this project is to save developer time and allow anyone to sketch an application and turn it into code.

I. LITERATURE REVIEW

The very first step in a project is the planning stage and therefore and user interface design must be ready to do that part. Designers mostly face challenges while doing this step as this is time consuming and expensive. The clients do several iterations with the design in their mind and hence most of the time of the developers I spent in changes it. In this project the authors solved this problem. Built an application that converts sketches directly into code. This project uses computer vision techniques to solve this problem and train a model to generate this UI interface code. For training, datasets were obtained manually, and the authors used data augmentation on the datasets. This project uses both the Tensorflow object detection algorithm and the Yolo algorithm to detect objects in sketches, such as buttons and text, and translate them into a programming language. This project also uses other computer vision techniques to recognize colors and handwritten text in user interface sketches. Users simply take a picture of their sketch and upload it to the application. The algorithm works on the backend and makes the code for this sketch available for users to download. The goal of this project is to save developer time and allow anyone to sketch an application and turn it into code.

Professors at the University of California and Carnegie Mellon University implemented SILK (Sketching Interfaces Like Krazy), this is a tool which is very much like paper based sketches as it includes all the advantages of paper-based sketching and much more. With SIILK users can provide much more accurate designs than the paper-based one as it includes materials such as an electronic pad and a pointer. This paper also discusses the abilities of SILK which include the drag and drop capability, the user can simply make up a container and drag and drop the elements in it. So, with this the user can see the whole working of the components hence, it is much easier than the past one but still it has limitations [4]

Through a set of studies, we got to know that web designers make designs at several levels of refinement, such as they can make sketches at the storyboard level or even the page wise level. The past paper-based sketching tools could not give this functionality to the users hence, DENIM was made. DENIM was made to assist web site designers at the early stage of design, and it also provides the ability to increase the size of the design which means that the designer can first make up a rough design on a page and if he/she get it approved so instead of making a whole new one on a storyboard they can simply increase its size. The authors named the system DENIM, which stands for Design Environment for Navigation and Information Models. DENIM is not made to output complete web site designs which means that it will not produce HTML and CSS pages. With this in mind, we just can design a web page or even a storyboard but cannot take all of that and build it up in any other interface without doing everything from the start. DENIM was built on JAVA 2 and with SATIN as the tool kit. With SATIN, it provided the pen-based system for making the designs. You could easily draw the designs with the pen architecture and label those designs. The label would be consistent throughout all the zoom levels as it needs to be read at all levels. There are several limitations of DENIM but what the authors wanted from it, it produces that.[5]

Single image super-resolution (SR) is a technique used in computer vision and is a really important technique. This is used to convert a low-resolution image into a high-resolution image. With this we know that there must be several stages in its pipeline. Firstly, the overlapping patches from the low-resolution image cropped in order to make it of the same shape as the high resolution image. It performs three techniques; subtraction, mean and normalization. With subtracting it can make the pixels as same the new image. With normalization, it makes the image easy to be input in the pipeline as the pipeline requires some specific inputs. So, then we start taking the weighted average All the other steps have not be optimized hence discussing those is not relevant in this paper. The paper has a vast number of techniques which they can improve. [3]

Video classification is used to produce accurate labels for a video given in the input. So how does it work? The labels are assigned frame wise which means that the classifier looks at the frames of thee video and then inputs those frames into the model. With this the classifier gives out a label for that frame. To make thee video classifier more accurate we can produce more than one label. So there can be more than one label for a particular video frame hence, more than one label for a whole video. [2]

Now there are several advancements in the field of computer vision among which is recognizing the description of the image in English. This application can be very much useful for the visually impaired. Till now image classification and object recognition applications have a very high scope in the artificial intelligence market, as there are several applications which can be made using those. Still, machine translation has been really important nowadays, people want everything to be translated in their native language. Even for this application, the visually impaired would want it in their language. Means that the model should describe the image in their native language. This all is possible and not at all complex with recurrent neural networks (RNN). We can achieve machine translation with this technique. This technique arranges the input sentence in a very rich vector which then can be provided to the model, and it will generate the translation into another language [6]

The user gives the graphical user interface design to the application, and it generate the code for it. How does it work? The application has a deep learning model embedded in it. They have already trained the model on several images that a person can provide to them. Then those images are also used to code the design hence they also needed corresponding codes for each widget that they trained the model on. They used text, buttons and several other widgets in their project. Hence, they also coded all the widgets in React Native. The images were first taken as an input to the model also they were first labeled in order for the model to be able to work on them. If it was a text widget then the image must be labeled as text. If they used tensorflow in it then they must produce json files for each image. Then only they could feed it into the model. This paper has used a deep learning and computer vision technique to build this project. So there are three things with every widget. The design of that widget, the label of that widget and the corresponding code of that widget. The design is important as this will detect how much similar is it to the design user inputs. The label is important as it will see which class the model will output it to. The code is important as after the classification, the code must be written automatically. The dimensions will just change but the object will remain what it was detected to. [1]

II. MODEL ARCHITECHTURE

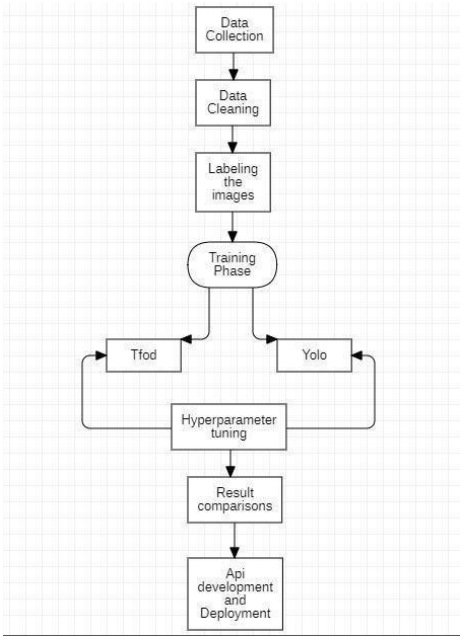

Fig. 1.: Shows the flow of application

A. Data Collection

In our project, we used the idea to draw the sketches of the UI on paper and we take the pictures of those sketches and used two approaches for the detection of the UI which will be discussed further. In our first phase, we targeted five widgets for UI detection, we draw each widget on paper with different combinations, then took a picture of each page and used algorithms for the detection of each image. Our algorithm gives the prediction and generates a code of the corresponding UI in dart language. We have collected images of hand-drawing widgets on paper and applied the data augmentation to them. The purpose of this project is to draw the images of the UI on paper and take a picture of it, which is the input of our model, and is to generate the code of the UI in a dart language which is used for mobile, web, and desktop applications, the code is the output of our model. The output of our model contains 2 images, the first image containing the code of the UI and the second output is the image containing the bounding box around each widget which gives a percentage of the occurrence of each widget. The purpose of our project is to solve the problem of hiring a separate UI designer. Our model is giving as accurate a result as possible.

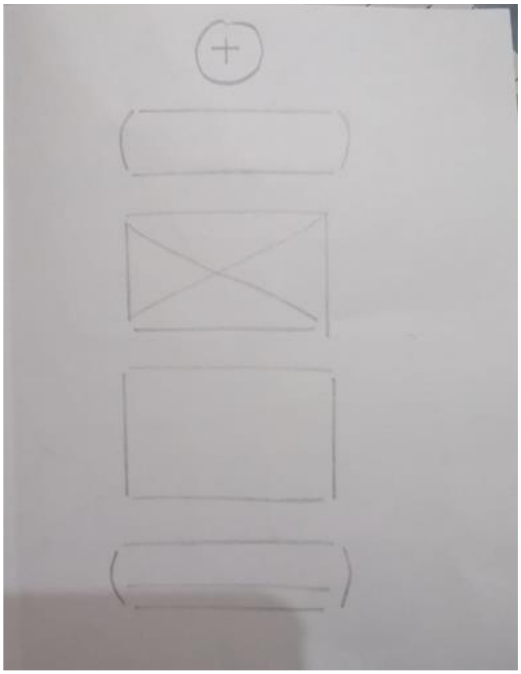

Fig. 2.: Sample dataset

B. Model Training

For the training of the dataset, we have used two approaches YOLO and Tensorflow object detection. YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time and take input labels. Object Detection using Tensorflow is a computer vision technique. As the name suggests, it helps us in detecting, locating, and tracking an object from an image or a video. As we train our models on both YOLO and Tensorflow object detection we got accurate results with both. Both the models gave accurate predictions. All the widgets such as containers, button, field, image and floating action button we accurately represented. We also trained out model on OCR for text and color detections and the models gave accurate results. As we give image input with widgets along with text and colors we get similar output.

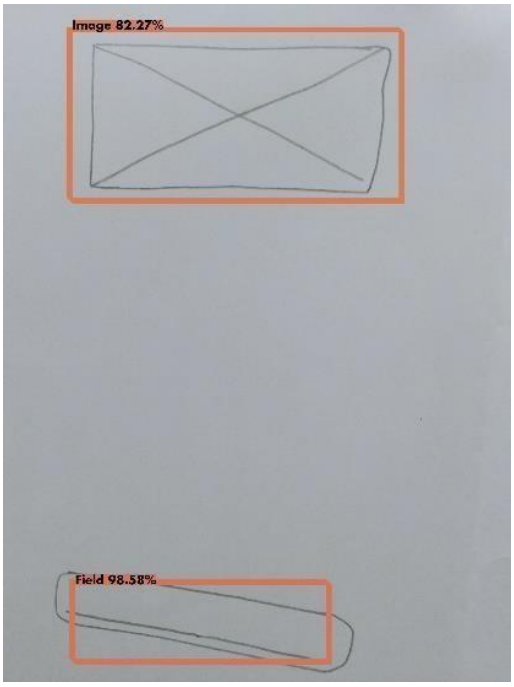

Fig. 3.: YOLO prediction

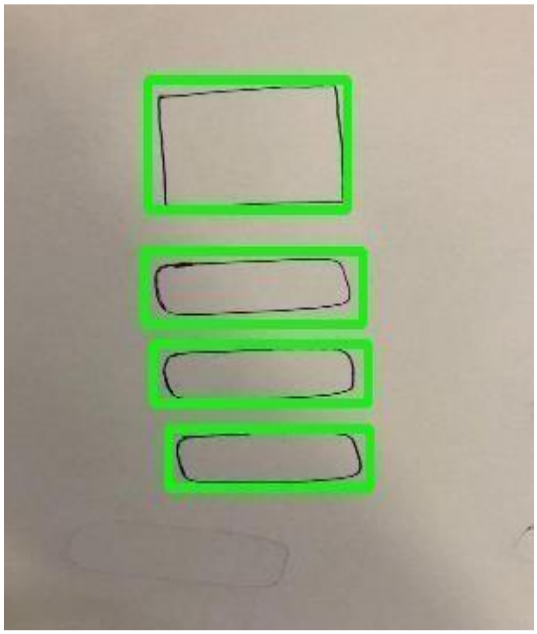

Fig. 4.: Tfod prediction

C. Text and Color detection

In order to detect color of widget, we train our model with some widgets that are labeled with colors name. We choose only green, red and blue colors initially. The backend api returns encoded color value which then converted into colors on frontend.

Same strategy we use for text as we use in color. All 26 alphabets are encoded in numbers at backend. Api will return encoded number that represents some text. Now this encoded-text would be converted into Text at frontend. For this we have used OCR.

OCR (scanning of documents) is the use of technology to identify printed or manuscript text figures inside digital images of tangible documents, such as a leaf through the paper document. The basic process of OCR includes examining the manual of a document.

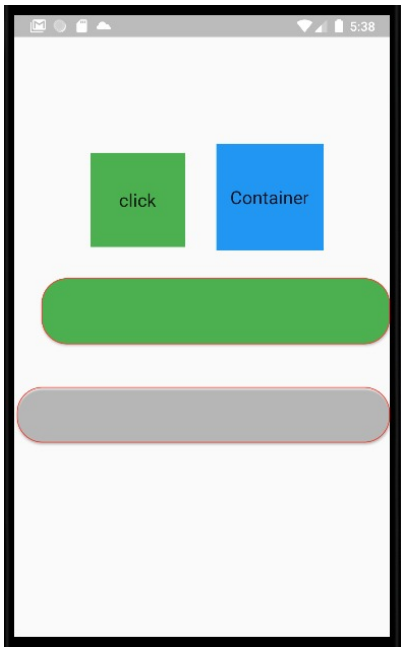

Fig. 5.: Text and color detection

D. API integration

The models and file labels that are generated have been integrated into an API.The model generated from TensorFlow has been integrated into Fast API. Now we have 2 endpoints of API, one endpoint returns images with bounding boxes (detection) upon giving images as input. Other endpoint returns JSON with X Y coordinates and class

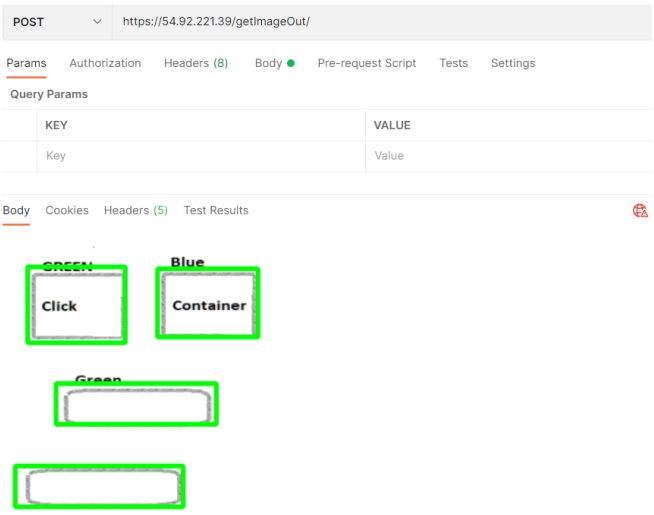

Fig. 6.: API returning bounding boxes

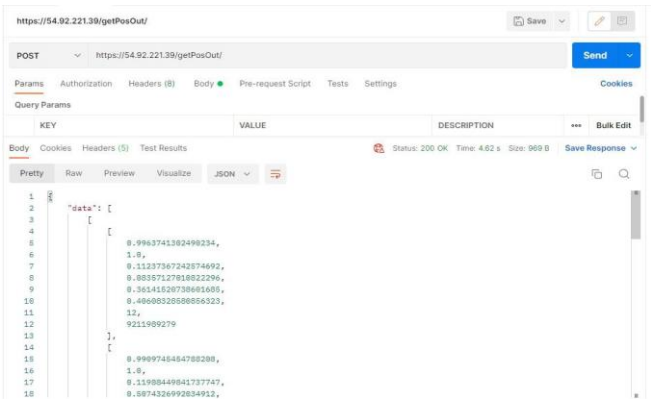

Fig. 7.: API returning JSON

E. API Deployment

REST api and models both are deployed on AWS. We can access it through Linux commands.

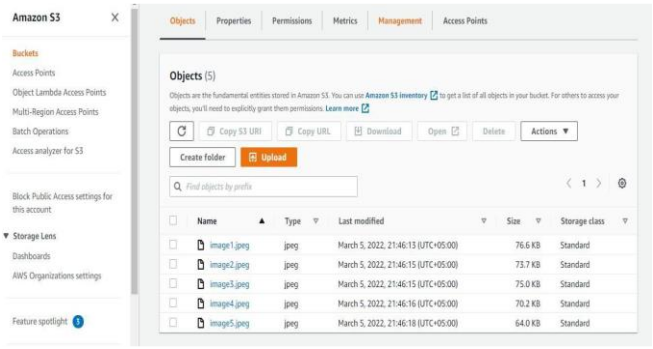

Fig. 8.: API deployed on AWS

III. RESULTS

After training the mode we tested it on 10 images and got accurate result on 8 images. We are successfully able to detect text as well as colors. But there were some issue like if the text was too big for the widget then that widget didn't appear at output as if the widget is not correctly drawn then we were not getting that at out.

IV. CONCLUSIONS

Sketch2code facilitate the developer by automating the initial stage of software development life cycle i.e. the user can draw the sketch of UI that he wants and the application will do the rest i.e. generating the corresponding UI and code of the sketch. The plus point is that the code is generated in dart language which is used to make cross-platform mobile, web and desktop applications.

V. ACKNOWLEDGEMENTS

The successful completion of the paper would only be possible with the constant inspiration and uplifting of the people who helped us achieve this milestone. First, we would thank Almighty Allah for guiding us on the right path and helping us wherever we were stuck. The role of our parents was also very important and cheering for us throughout this journey. We would like to express our appreciation towards our internal advisor Dr. Maria Waqas, Assistant Professor, Department of Computer and Information System Engineering for giving us the opportunity to work on this idea andguiding us regarding the project, and taking out time from her busy schedule to assist us in this long journey.

REFERENCES

[1] Alex Robinson. (2019). Sketch2code: Generating a website from a paper mockup.

[2] Balakrishnan Varadarajan. (2015). Efficient Large Scale Video Classification.

[3] Chao Dong. (2015). Image Super- Resolution Using Deep Convolutional Networks.

[4] J.A. Landay and B.A Mayers. (2001). Sketching interfaces: toward more humaninterface design.

[5] james Lin. (2000). DENIM: Finding a Tighter Fit Between Tools and Practice forWeb Site Design.

[6] Oriol Vinyals. (2014). Show and Tell: A Neural Image Caption Generator.