Sketch2Code: HTML code generation from sketch

Darji Devam, Dalesh N, Faizan Rashid, Kushagradheer S, Dr. Geetha S

Bnm Institute of Technology
Bnm Institute of Technology
Faizan Rashid Hakeem

Abstract

In the initial phases of program development, user interface (UI) prototyping is a crucial step. A Graphical User Interface (UI) designer's uninspired yet time-consuming duty is to turn designs of a UI into a programmed UI application. This process will be significantly sped up by an automated system that can take the place of humans in the simple execution of UI ideas. The works that support such a system strongly emphasize using UI wireframes as input rather than manually produced sketches. Using a Deep Neural Network that has been trained on a database of such sketches, we offer a novel technique in this research for UI element recognition in input sketches. The specific visual identification task of object detection in sketches is addressed specifically by our deep neural network model. The output of the network is a platform-independent UI representation object. A dictionary of key-value pairs is the object used to represent user interface elements and the properties that go along with them.Our UI parser uses this as input and generates code for many platforms. Because of its inherent platform neutrality, the model can train once and produce UI prototypes for other platforms. This two-step strategy outperforms existing approaches by producing accurate results quickly (on average, 129 ms), without the requirement for two trained models.

***Keywords*: Sketches, Yolo V5, User-Interface, PyTorch, HTML, Dataset, Machine Learning**

Introduction

The component of an application known as the user interface determines how a user interacts with it and makes it possible for them to use its features. Developers and designers have embraced sketching as a tool for quickly sharing their ideas or opinions with stakeholders. UI design requires the quick execution of UI Sketches as review-ready prototypes. Developers spend a lot of time creating a prototype in order to collect feedback from stakeholders. After this stage, the prototype's code is removed, and the application is constructed via the typical phases of application development. Developers then turn these sketches into UIs that are coded.They are unable to work on the application or website's distinctive features because of this time-consuming process. Machine learning-based code generation from sketches is a relatively recent area of study. Interpreting sketches into images by a machine is a computer vision challenge since it involves a system making deductions from sketches, understanding them, and extrapolating logical information from them. Eyiokur et al. brought this issue to light and showed how different combinations of techniques are required to obtain high accuracy. Computer vision has come a long way since its debut. The main source of data for early research studies in the field of computer vision was computer-aided design (CAD). The transformation of architectural pictures using CAD technologies produced excellent results. Convolutional neural networks have evolved, and deep neural networks (DNN) have recently become quite popular (CNN).Object recognition in images has been a common problem in computer vision, and CNN has emerged as the most effective solution. According to Szegedy et al., DNNs can be utilized for object detection and precise localization of items belonging to various categories. The results showed that basic formulation could result in powerful results when combined with a "multi-scale coarse-to-fine technique." Wang et al. classified photos using the Convolutional Neural Network rather than just concentrating on feature extraction. The results showed how a multilevel convolutional system may be used to process a source image in order to characterize it from a new perspective. As shown by Chao Ma et al., using many convolutional layers can increase accuracy. Hierarchical features of convolutional layers can be employed to represent the original item. Erhan et al. suggested using a saliency-inspired neural network to forecast class-independent bounding boxes and assign each box a score that represents the possibility that it will include any valued objects. The majority of the study has focused on using actual photographs to detect objects. In contrast to when the image is a hand-drawn sketch, the models that have been developed in this way are extremely effective and educated in their subject. We looked into a variety of models and evaluated how well their structure and methodology may fit our use case.

Literature Survey

Machines have significantly advanced humankind's technical state. For everyday tasks, from performing our rudimentary computations to a more effective AI However, the latter is more adaptable and provides the opportunity to navigate difficult circumstances, therefore merging both approaches into a single system would result in the best of both worlds. One of the areas of scientific research that is now growing the fastest is intelligent systems. They can replace humans in numerous sectors thanks to their talents. Applications include monitoring, planetary exploration, reconnaissance, industrial automation, building, amusement; tour guides for museums, personal services, transportation, and so forth.Numerous other industrial and non-industrial applications, among others. The majority of these are already on the market. The use of the OpenCV libraries for the purpose of detecting license plates and recognizing vehicle identification numbers is one such use. In this application, we essentially identify the numbers and characters using license plates. Our concept is easier to put into practice because of a similar idea.Currently, users must create HTML code to structure the webpage's elements. This requires a lot of repetitive labor and takes up valuable user time. Users typically save a copy of the pre-existing boilerplate code for reuse when the structure of web pages is the

same.Even though the boilerplate code for HTML elements is often the same, it varies for web pages. The labor is rendered unnecessary in these circumstances. To make life easier for the community of web developers, this procedure can be automated. We suggest a machine learning model make this process simple. It will be taught to recognize particular symbols and forms as elements and to read messages from wireframes. Through the web application, wireframe images will be input into the model. The goal is to receive data and use an open-source computer vision library to identify each wireframe element.

Proposed Methodology

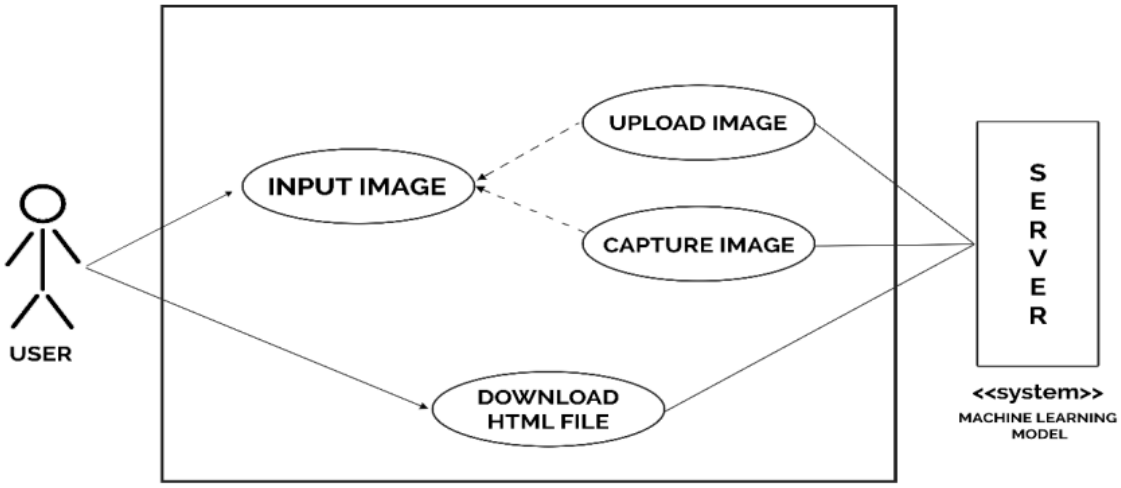
Algorithm Input: For training and testing, the model will need a dataset with a variety of wireframe drawing samples. The input dataset will include various artists' sketches of the web user interface's component parts. The components must be drawn in accordance with a predetermined set of guidelines and symbols. This will make it simpler to identify components in the photos.

Image Analysis: After the input has been given, it is time to extract useful information from the wireframe, but first the image needs to be pre-processed. This involves turning the image to grayscale, performing noise reduction and edge detection, as well as other operations to make the Pattern recognition stage of HTML element detection easier. Several Python image processing libraries will be used to achieve this. Python modules like the Python Imaging Library (PIL), which has the Image Ops module with easy-to-use image processing operations that may be applied to the image, can be utilized in the image analysis section. The wireframe sketch will be processed using Yolo V5, which will also be used to identify labels, shapes, symbols, and text.Tesseract OCR can be used to recognize, retrieve, and include text from the wireframe image into the HTML code output.

Pattern Recognition: Once the image has undergone pre-processing and has been examined for fundamental image analysis elements. From the wireframe designs of the training dataset, pattern recognition will be used to identify patterns for all HTML elements and web components. For each type of web component, a distinct symbol or shape will be drawn. For instance, a circle with dots for radio buttons, a plain rectangle for text boxes, and a button with text in the midst of a shaded rectangle. To identify these items in accordance with these patterns, a unique library will be created and used. The identified HTML element will be surrounded by a box. To ensure that the arrangement of the web elements is done appropriately, a layout technique can be employed to create a grid structure that accommodates all the boxes that have been formed.

Processing: The developer will create an internal library that includes specific pre-set data and pattern sets that the model will need in order to analyze the data. A fundamental collection of structures and patterns will be defined in our model.

Flowchart and Steps Explanation



1) We will be concentrating on the object detection models of YOLOv5 to detect all the attributes and match them with the data fed. If the accuracy will be greater than the threshold then it will be detected, otherwise will show a failed detection message. The labels, text fields, buttons, checkboxes, lists, and other multiple setups will be fed into it before making an analysis. Tesseract and Ease text detection libraries, which contain the setup of a) Pre-processing the image using HED of any edge detector and various perspective transformations, will be used to determine the functionality of the other labels and buttons. b) Using the Pytesseract and East Text Detector to read the image's text regions. Results can be obtained by directly passing a picture through the tesseract. However, preparing the image is where the magic happens, and by doing so, we may significantly enhance the final product. To obtain a clearer image, it will be followed by the noise reduction phase. The next step is contour detection, which separates large, small, and other contours. After the pre-processing phase is complete, we will now move on to label detection.
2) We must sort the detected attributes from left to right and top to bottom to obtain accurate results. After sorting we will get a 2-D array in which each 1-D array represents a row in design. This sorted 2-D array is passed through a function to obtain the final HTML code as output.

Proposed Implementation

Dataset:

Attributes recognized by model

Button, Carousel, Image, Heading, Pagination, Select, Table, Text-Box, Paragraph

Proposed algorithms

YoloV5 object detection

Steps of execution

1) We will first train the small YOLOv5 model, then train a medium model, and finally we will examine the output variance between the two.
2) We will do inference in each of the above mentioned scenarios, comparing the FPS and mAP metrics as we go.
3) HTML code will be generated according to the inference of YOLOv5 model.

DDFK (Modified YoloV5 by our team)s

Steps of execution

- We will first train the small YOLOv5 model, then train a medium model, and finally we will examine the output variance between the two.
- We will do inference in each of the above mentioned scenarios, comparing the FPS and mAP metrics as we go.
- Output gathered from the above step will be passed through a sorting algorithm, which will sort the bounding boxes from left to right and top to bottom.
- HTML code will be generated after the bounding boxes is sorted and more accurate results will be seen.

Results

Fig 1. Hand-drawn sketch

Fig 2.

Recognition of attributes by model with labels

Fig 3. HTML code output

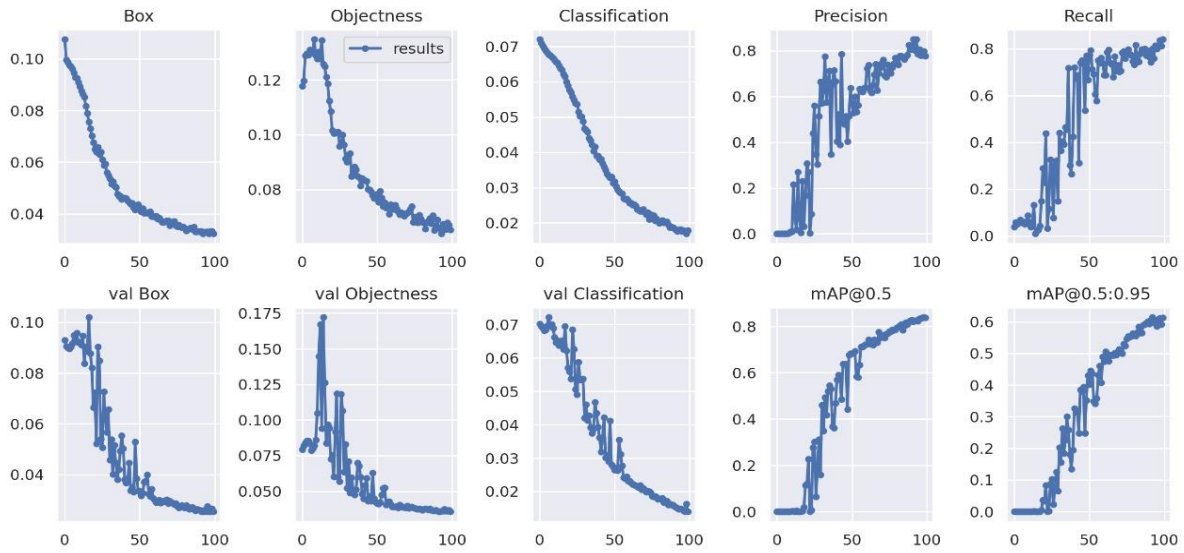


Fig 4. Statistics of model

Conclusion

The outstanding work, Code-AI, demonstrates that we can train a neural network to identify user interface elements in hand-drawn sketches. Performance, which is affected by the training set, can be improved by providing more labelled example sketches. During the analysis, we evaluated the model on a set of different images. The model learns the shape of the UI elements, and after fixing the overlapping elements, it outputs the final UI representation object allowing the UI parser to generate the interface. Platform-specific user interface.

References

[1] B. Myers, S. E. Hudson, R. Pausch, and R. Pausch, "Past, present, and future of user interface software tools," ACM Trans. Comput.-Hum. Interact., vol. 7, no. 1, pp. 3–28, Mar. 2000.
[2] F. I. Eyiokur, D. Yaman, and H. K. Ekenel, "Sketch classification with deep learning models," in 2018 26th Signal Processing and Communications Applications Conference (SIU). IEEE, 2018.
[3] I. M. Rian, M. Sassone, and S. Asayama, "From fractal geometry to architecture: Designing a grid-shell-like structure using the Takagi–Landsberg surface," Computer-Aided Design, vol. 98, pp. 40–53, 2018.
[4] N. O. S¨onmez, "A review of the use of examples for automating architectural design tasks," Computer-Aided Design, vol. 96, pp. 13–30, 2018.
[5] S. Modi, M. Tiwari, Y. Lin, and W. Zhang, "On the architecture of a human-centered cad agent system," Computer-Aided Design, vol. 43, no. 2, pp. 170–179, 2011.
[6] M. Hablicsek, M. Akbarzadeh, and Y. Guo, "Algebraic 3d graphic statics:Reciprocal constructions," Computer-Aided Design, vol. 108,pp. 30–41, 2019.
[7] S. He and N. Pugeault, "Deep saliency: What is learnt by a deep network about saliency?" arXivpreprint arXiv:1801.04261, 2018.
[8] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE international conference on computervision, 2015, pp. 1440–1448.
[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, "Focal loss for dense object detection," IEEE transactions on pattern analysis and machine intelligence, 2018.
[10]Aparna H., Dr AbhijitJoshi , "Generation of Web Pages from Document Image", IJAIS Proceedings on International Conference andworkshop on Advanced Computing 2014.
[11].AnujaNagare, "Licence Plate Character Recognition System Using Neural Network", International journal of Computer Application, Volume 25– No.10, July 2011
[12]RuhinaKarani, Dr.TanujaSarode, kekre H B, "A deviant transform based approach for color image registration", IEEE Communication, Information & Computing Technology (ICCICT), 2012
[13]RuhinaKarani, Dr.TanujaSarode "Image Registration using Discrete Cosine Transform and Normalized Cross Correlation", International Conference & Workshop on Recent Trends in Technology, (TCET) Proceedings published in International Journal of Computer Applications (IJCA) 2012
[14] L. G. Brown, "A survey of image registration techniques," ACM Comput.Surv., vol. 24, no. 4, pp. 325–376,.Jawad AI K.,Jiang, Haikal EI Abed "Word-based Handwritten Arabic Scripts Recognition using DCT Features and Neural Network Classifier", 2008 5th International Multi-Conference on Systems, Signals and Devices978-1-4244-2206-7/08 2008 IEEE Dec. 1992
[15] V. Kalaichelvi, Ahammed Shamir Ali ,"Application of Neural Networks in Character Recognition" International Journal of Computer Applications, by IJCA Journal 2012
[16] "Introduction-to-Artificial-Neural-Systems", p*ublished by West, by J.M.Zurada 1992*