International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

UI Code Generation using Deep learning

Aditi Deolekar[1], Kimaya Dhanawade[2], Rahul Chavan[3], Sayali Bhambure[4]

[5]Dr. Prashant Nitnaware, Professor, Department of Computer Engineering, PCE, Navi Mumbai, India - 410206

Abstract - *Implementation of GUI code in any project is a tedious, time-consuming job for front-end developers. It takes most of the time from effective logic building and working on the actual functionality of the code implemented in a project. Web technologies or languages used for a project are very specific to each target runtime system. Hence the work can be repeated several times.In our project, we describe a model that is trained end-to-end with gradient descent and learns Sequence Modeling and visual feature extraction at the same time to generate variable-length strings of tokens from a single GUI image as input (to generate code from single image input). We are proposing our project model based on CNN (Convolutional Neural Network) and GRU (Gated recurrent units). Using Deep Learning Techniques, the model takes input as a single GUI screenshot, breaks it down into various tokens to give us the ultimate code for it. We can even train the model on different data sets for different effective output codes.*

Keywords - HTML, Deep Learning, Dataset, Convolution Neural Network, Gated Recurrent Unit.

1.INTRODUCTION

The very first step in creating an application or website is to sketch a wireframe or to make a graphical user interface screenshot. Designers face a challenge when converting their wireframe or GUI into code, this work often consumes time for the developer and therefore increases the cost.

The process of implementing a Graphical User Interface (GUI) mockup created by a designer and converting directly into a website is the responsibility of developers. Most present-day user-facing software programming applications are Graphical User Interface (GUI) driven, and depend on alluring User Interface (UI). But implementing GUI code is, however, time-consuming and prevents developers from dedicating the majority of their time to implementing the actual functionality and logic of the software they are building. The project is from a single GUI image as input to generate computer UI code, using Deep Learning Techniques. To train the model on different data sets for different effective output codes. We want to build a neural network that will be generating HTML/CSS markup that corresponds to a users screenshot. When you train the neural network, you give it many screenshots with matching HTML. It learns by predicting the matching HTML markup tags one after the another. When it predicts the next markup tag, it receives the screenshot and all the correct markup tags until that point.

2. LITERATURE REVIEW

Present-day, front-end developers work so hard for a perfect GUI interface. Avoid the frustrations that front-end engineers and designers experience when creating precise GUIs; this highlights the need for more effective resolution in web page design. In this study, to automatically produce the HTML code for the mockup of a website page an approach has been created. It is planned to recognise the elements created in the mock-up images and encode them in accordance with the website page hierarchy. To train, the deep neural network model, which includes Convolution Neural Networks (CNN), is utilized to train the images present on the data sets.

A recent example is pix2code, an approach based on Convolutional and Recurrent Neural Networks which allows the production of computer tokens from a single Graphical User Interface screenshot as input but their Model was trained on a relatively small dataset, hence accuracy was less. Whereas, Sketch2code used classical wireframe techniques and deep learning techniques code was generated by pre-processing and segmentation but it also has moderate results since input is based on the camera of the device.

Another example is DeepCoder, a system able to generate computer programs by leveraging statistical predictions to augment traditional search techniques. The technique reduces the runtime of a wide variety of IPS baselines by 1-3 orders of magnitude.Several problems in real online programming challenges that can be solved with a program in this proposed language. But the problems which are solved by the model are relatively simpler than any other competitive exams. The input-output examples become less informative as the DSL becomes more complex.

In this paper, Graves explains how to use a Recurrent Neural Network to approximate probability distribution function (PDF). The paper proposes using LSTM cells in order to train the RNN to remember information from the distant past. With that modification, the PDF of the next value of the sequence can be approximated as a function of the current value of the sequence and the value of the RNN's current hidden state. The model fails to train on

more complicated datasets, due to the more complex and diverse nature of each image.

In Improving pix2code based Bi-directional LSTM by Yanbin Liu, Qidi Hu, Kunxian Shu where model is optimized by Bidirectional LSTM, which allows the output layer to get complete context of past and future information for each point in the input sequence.. Decoder exploits the advantages of CNN in feature extraction combined with the advantages of BLSTM in processing sequence problems to automatically generate code.The model's transforming accuracy in the test set has been significantly improved reaching 85%. But since, The model uses Bidirectional LSTM, hence the training time of the model is more.

3. PROPOSED WORK

Generating computer code from a GUI image is the same as getting an English text from a photograph. Long Short-Term Memory(LSTM) is used to predict the sequences. Instead of this, GRU(Gated recurrent unit) can be used in the system. The GRU is the same as a long short-term memory (LSTM) with a forget gate, but it has fewer parameters than LSTM, as it lacks an output gate. The key difference between GRU and LSTM is that GRU has two gates that are reset and update while LSTM has three gates that are input, output, and forget. GRU is less complex than LSTM because it has fewer gates. GRUs have been shown to display better performance on certain smaller and less frequent datasets. Hence, this can increase our accuracy level in the proposed system. GRU is used for processing, analyzing the sequence of training data and Convolutional Neural Network (CNN) which is adequate for image recognition and analyzing segmentation. Both the techniques are trained with a dataset and give us target output code. Hence variable-length strings of tokens from pixel values are produced using these techniques. A full functional automated UI code generation system would be useful in GUI code Generation and Software UI Prototyping.

3.1 System Architecture

A system having deep neural networks can learn latent variables which describe the objects in an image and their relationships with the corresponding variable-length textual descriptions. To acquire this, our system is segregated into three sub-models mainly the computer vision model, language model and code generation.

Fig.1 Flowchart of the architecture.

A. Input:
As an input, a dataset consists of GUI images, and some contextual code data is provided to the system. Data is not combined but is linked with each other and trained on two different models to give the optimum result. Deep neural networks can learn latently.

B. Computer Vision Model:
Convolutional Neural Network helps to analyze visuals. In the Computer Vision model, CNN is used to perform an unsupervised feature which is learned by mapping an input image to a learned fixed-length vector, thus this acts as an encoder. Pixel values of an image are normalized before giving to CNN, hence the initial output vector is generated.

C. Language Model:
Gated recurrent units consist of an update gate that decides how much past activation information is required for the next activation step and a reset gate that decides what information to forget. In the Language model, GRU trains on sequences of strings to generate a fixed-length vector. Domain-specific language (DSL) to represent HTML. Due to the high complexity of native web code, domain-specific languages (DSLs) are generally designed. DSL contains the syntax and the semantics which were needed for modeling, and DSL does not use loop statements or control statements, so it greatly shortens the vocabulary. A language model can achieve token level language modeling with a discrete input by using one hot encoded vector; eliminating the need for word embedding techniques such as word2vec that can bring in expensive computing. Traditional RNN architectures suffer from vanishing and exploding gradients which prevent them from being able to model such relationships between the data points spread out in time series. The Long Short-Term Memory(LSTM) neural engineering is used to address this very issue.

D. Code Generation:
Training: As shown in fig, vectorial represented image and tokenized vectors trained by respective models linked together in vector rt and fed into the second GRU model to learn object representation in GUI image. The softmax layer is used to perform multi-class classification. Softmax layer calculates the probability for each possible output vector that is a DSL token which in this case is HTML/CSS code.

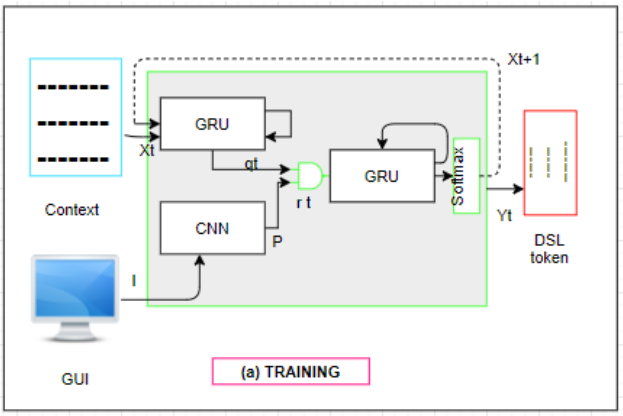

Fig. 2 Proposed system architecture

Sampling: During sampling, the input content is updated for each prediction to contain the last predicted token that is xt is set to xt+1, which gets output as yt. With xt+1 the expected token, and yet the predicted token. Using compiler techniques the resulting sequence of DSL tokens is compiled to the desired target language.

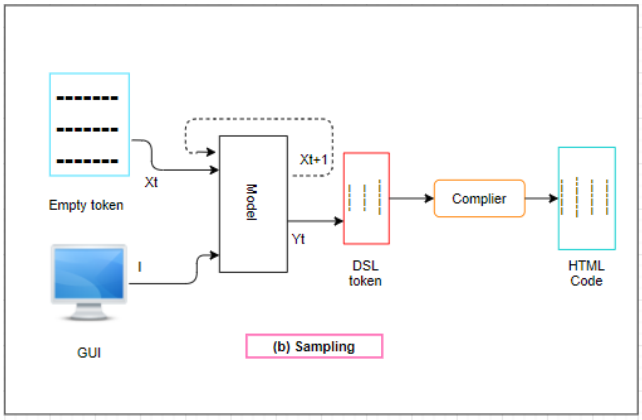

Fig. 4 Proposed system architecture

The model is optimized end-to-end and hence the loss is minimized concerning all the parameters including all layers in the CNN-based vision model and all layers in both GRU-based models.

4. REQUIREMENT ANALYSIS:

4.1 Dataset:

For training we are using the same data set which is used by the pix2code model. The column Synthesizable refers to the maximum number of unique GUI configurations that can be synthesized using our stochastic user interface generator. The number of synthesised instances is indicated by the column Instances(GUI screenshot, GUI code) file pairs. The number of distinct image-sequence pairs is indicated in the column Samples. In fact, training and sampling are done one token at a time by giving input to the model with an image and a sequence of tokens obtained with the sliding window of fixed size T. The total number of training samples depends on the total number of tokens present in the DSL files and the size of the sliding window. Synthesize GUIs written in our DSL which is then given as input to the compiler to get our desired target language to be rendered.

Dataset type	Synthesizable	Training set		Test set	
		Instances	Samples	Instances	Samples
web-based UI (HTML/CSS)	31×10^4	1500	143850	250	24108

Fig.3. Dataset

4.2 Software

Operating systems of Windows 10 along with Python which is a largely used programming language for deep learning models is required. Keras is a python built neural network library which is required to build models in Tensorflow 3.3.0.10. Other libraries like NumPy are required for better computational performance. A stable cloud environment is required such as Google collab.

4.3 Hardware

Training a Deep Learning model requires 2GHz of intel processor, a hard disk drive of 180 GB, and 16 GB RAM. Computers with the graphics card, GPUs can train models faster. But for our proposed system, building models on cloud service is more preferable and what has been used.

5. CONCLUSION

Transforming website mock-ups into mark-up code with less time along with minimal cost has been a crucial point. In this paper, we developed an approach that accepts web page GUI screenshots, processes them, and generates structured HTML code. A dataset comprising pictures, including different mockups of web page structures were utilized to train the GRU model. GRU trains quickly. The accuracy of our model is increased and also does not comprise many limitations and is prepared on an earlier dataset. In fact, one could imagine crawling the World Wide Web of HTML/CSS code associated with screenshots within a few minutes.

More future work and knowledge is needed on further improving the performance of the code generation. There is a need in the industry, in our everyday life for such applications because every company wants to make their tedious work easy and efficient. This research proposes a system which automatically converts the input image data into html website. A direction for future research is to carry out the implementation of the system and check execution by applying proposed approaches to various benchmark data sets. Comparing performance of different

classification methods to find the best one for our proposed code generation could be another future research direction. However, by increasing the data set and the training period we can increase the accuracy of our model for faster and accurate output. Introducing the concept of transformer to replace it with RNN, LSTM is an area of great research for more effective models.

ACKNOWLEDGEMENT

We would like to thank our guide and mentor Dr.Prashant Nitnaware who mentored and supported us throughout our "UI Code Generation using Deep learning" project based on the domain "Artificial Intelligence and Machine Learning".

We would also like to thank our principal Dr. Sandeep Joshi and the Head Of Computer Department Dr. Sharvari Govilkar for giving us an opportunity to understand and implement deep concepts of Artificial Intelligence, Machine Learning, and Deep learning providing us with all the facilities and equipment required for this project.

REFERENCES

[1] Tony Beltramelli UIzard Technologies Copenhagen, Denmark, "pix2code: Generating Code from a Graphical User Interface Screenshot" Proc.ACM SIGCHI Symp. Eng. Interact. Comput. Syst, June2017.

[2] Alexander Robinson, University of Bristol, "sketch2code: Generating a website from a paper mockup," Department of Computer Science, May 2018.

[3] Matej Balog, Department of Engineering University of Cambridge and Alexander L. Gaunt, Marc Brockschmidt, Sebastian Nowozin, Daniel Tarlow, Microsoft Research, "DEEPCODER: LEARNING TO WRITE PROGRAMS," ICLR 2017.

[4] Yanbin Liu, Qidi Hu, Kunxian Shu, "Improving pix2code based Bi-directional LSTM" IEEE International Conference on Automation, Electronics and Electrical Engineering, 2018.

[5] Alex Graves."Generating Sequences With Recurrent Neural Networks" Department of Computer Science University of Toronto, June 2014