



# A SURVEY ON AUTOMATIC CODE GENERATION USING IMAGE MOCK-UPS

Harshitha. S, Devi Kannan

<sup>1,2</sup>Atria Institute of Technology, Bangalore, Karnataka, India

**Abstract:** The idea is to use mock-ups as a way of designing the user interface of a complex program. The most complicated part of the software project is arguably writing the code to model the mock-up. So rather than spending time coming up with an algorithm to take care of the details, we use automatic code generation based on an action, objects relation, an optional default screen view. Deep learning models which transform image mock-ups of graphical user interface (GUI) to a program. It taken place by four steps object detection, object cropping, object recognition, program builder. This are used to build the websites, mobile application and software for the particular use. While in the stage of object identification the cropped components are labeled by CNN trained model.

**Keywords – Object detection, Object recognition, GUI, Deep learning, CNN, Automatic code generation.**

## I. INTRODUCTION

Programming is a challenging occupation, but it also brings with it many opportunities. Recent developments in programming language syntax and computer hardware allow programmers to create large and complex websites. These websites are not merely static pages; they are interactive, dynamic documents that can be used for many purposes including customer support, financing, marketing and education -- just to name a few. The field of programming continues to evolve with interesting new projects every year. websites for programming is also called web development and design, as well as HTML or DHTML. It's one of the most important technologies in digital marketing because every brand has a website now. Website is a software product designed to enable people to communicate with one another. It allows law firm websites to be easily navigated, organized and updated. It also provides multiple ways of communication. All these aspects make websites today indispensable for the law firm's success. A website is a collection of documents, like text or pictures, that seen in browser.

The most important pages are called the home page, which is where most users arrive from when they visit your website. Other important pages include other parts of your site's navigation menu, contact information such as business hours and email address. A software system which generates

code is often called "code generator". Image mock-ups are used extensively to develop prototypes, test new styles and GUI layouts. A smart way to visualize design ideas, image mock-up is a powerful tool for communicating your ideas to customers. Image mock-ups are used to test the usability of an interface by showing users how it would appear in their real smartphone or tablet environments. They allow the developer to provide a realistic image that can't be replicated directly in the actual application, instead making it clear how users will interact with specific areas of the application such as menus and buttons.

The CNN method used to process image data is one of the most successful methods since the beginning of computer vision. It has been used in many applications such as feature detection, object segmentation and classification. CNNs are being used to support applications ranging from self-driving cars to robotics. CNNs are also used for pattern recognition, in which there are examples of neural networks recognizing images and other inputs. The accuracy of CNNs has improved greatly over time thanks to faster CPUs and GPUs, as well as advances in data science techniques like deep learning.

GUI is a graphical user interface for automatic code generation. GUI can be used to generate executable code from existing specifications and descriptions, or from scratch. It was originally developed at HP Labs and has been used in engineering design review, specification analysis, and prototype development projects across industries such as aerospace, healthcare, tele communications and software engineering. Automatic code generation is an automatic process that converts computer programs into machine language, a form that can be easily processed by a computer. Automatic code generation is generally used to create complex programs from simple ones by automating the process of translating one or more source programs into multiple target machine languages. Automatic code generation is the process of generating code from data and templates. In simple terms, it reduces your coding time by automating repetitive tasks.

## II. LITERATURE SURVEY

Moran et al. [1] proposed a GUI it is a software for prototyping mechanized for data handle process, and an application of this methodology is REDRAW implemented for Android. It will correctly determine the GUI elements



image and make the grading that is same as designer. IT will create the apps that are optically same as image monument and firmly effect trade system. Reverse Engineering Mobile Application User Interface as been introduced recognize components of UI through computer sight. REMAUI helps to remove the space between pictorial designer visionary picture and running UI program. It also included supplementary programme iOS and JavaScript. REMAUI will load the display to different implementation it was presented by Nguyen et al. [2]. Natarajan et al. [3] presented an utilizing computer vision techniques, the P2A tool creates animated mobile applications. The customer interface confluence in a supplication original program and it will prepare to act in accordance and to performed in phone is conclude by P2A from the designs of the screens for mobile applications. Among other things, implied mobile applications have in-screen animations and transitions between screens. Beltramelli et al. [4] Pix2code is a cutting-edge technique for create only one GUI code for a computer. We barely scraped the surface of what is feasible, even though our work shows the system's ability for brutalize the GUIs execution process. Our model includes the instructed portable data file and has a modest number of parameters. Instructing the lager prototype in substantially added facts for a greater amount of period could significantly raise the quality of the resulting code. Reiss et al. [5] presented the starting creator's representation for the alliance that can be closely matched or duplicated by a lot of example pictorial end user confluence for subsist unlocked origin apps. The methods required to turn a sketch into something that can be checked, to take out the right program for before existent program explore the machines, to modify code to a programme it will obey and execute in a contains the end user alliance, and finally it will enable the customer to combine with chosen conclusion of attentiveness. Karpathy et al. [6] presented a model with very few hardcoded assumptions that creates simple dialect explanation of picture area using feeble tag from a dataset of images and phrases. A cutting-edge ranking approach that integrated verbal and visual components using a shared multimodal embedding and a structure for a multisensory repeated sensual web that produces explanation of a vision data. The Multisensory RNN outclass recovery guidelines in one and the other full frame and region-level studies, which were used to evaluate its performance. Barnett et al. [7] Better code generation tools would be especially helpful for data-intensive apps since they frequently use boilerplate code. Developers of mobile apps must also make an effort to address a number of frequently incompatible issues. It offers RAPPT, an MDD-based tool that creates the scaffolding for an Android app, in order to allay these worries. RAPPT generates normal Android code while preserving all the framework's features. Gargenta et al. [8] proposed a mobile application for the Android operating system, developers can utilise the eclipse platform, Android ADT, and Android

SDK. These systems operate smoothly and provide a nice user interface. Today, Java on the Eclipse IDE is the platform of choice for developing software. In this setting, software engineering researchers have created programmes for a variety of objectives and then analysed them. The user experience offered by these programmes is excellent. Balog et al. [9] presented a methodology for IPS system improvement that turns input-output example cues into suggestions for where to look in programme space using neural networks. IPS system improvement framework utilising neural networks. Dai et al. [10] proposed a account diversity, naturalness, and semantic importance. To address the technological difficulties in end-to-end training. Ling et al. [11] presented Latent Prediction Network is a type of neural network architecture that effectively marginalises data across different predictors. A proactive representation of program creation it will integrate multiple indicator networks of duplicate keywords for the process with the property extent softmax to plan of action language-specific representation. Reed et al. [12] The model is capable of combining numerous realistic visual interpretations of a given text caption. The CUB's text to picture synthesis. We demonstrated how to separate style and content, as well as how to move bird poses and backgrounds from query images into text descriptions. Zhang et al. [13] presented for the creation of photorealistic photographs, use Stacked Generative Adversarial Networks (Stack GAN) with Conditioning Augmentation. The suggested approach breaks down text-to-image synthesis into a brand-new process of sketch-refinement. Moran et al. [14] introduced Gvt, a method for automatically identifying, categorising, and notifying design breaches in mobile apps, and carried out a thorough investigation to gauge the tool's effectiveness, usefulness, and potential for industrial use. It willdemonstrate that Gvt is practical, useful for developers, and relevant in industrial settings. Moran et al. [15] proposed CRASHSCOPE, a useful method for locating, documenting, and if a android operating system dash it will rerun. The strong method for structured survey it is impervious to crashes, effective to pop-up text generation, input, works in various platforms, it will also mimic. Deka et al. [16] presented Rico, the most comprehensive collection of a phone program plan to time, is presented. It was developed to serve quintuple kind of facts operation implementation of pattern explore, UI format creation, UI program creation, user interactivity modelling, user insight forecast. Moran et al. [17] Due to the required resources, the automatic input generation methods now used to Android operating system that are often not practicable to designer to utilize. Generally, they do not successfully exercise contextual characteristics due to instrumentation or platform dependencies. In this demo, we introduce CRASHSCOPE, a revolutionary automated tool designed to assist developers in mobile testing chores. Mao et al. [18] presented a novel multifunction exploration established operating system



methods and instruments. Sapienz to computerized Android operating system trial. Sapienzcarry multi-layered apparatus and residue appropriate and available for APK app. Menzies et al. [19] proposed an analysis to compare the baseline approach with the most advanced deep learning technique to forecast understanding component association in mass overspill. The performance of baseline techniques is enhanced through tuning. It performs in suggested CNN approach for the Word Embedding and SVM technique. Compared to complex deep learning, the control technique with framework adjustment will move in more speed. SVM tuning in this study is 84X faster than CNN approach. Dave et al. [20] proposed the development of intelligent automation, the problems it solves, the difficulties will still overcome, and various implementations using this mechanics contrast to one another. The website generating automation and recognise the unique strides they have taken. The potential for advancement is enormous, and it is advance in machine learning methods, possibilities for brilliant mechanization, particularly in web page designing area, will significantly expand. Machado et al. [21] proposed prototype form end user confluence evolution, user-centered creation, prototype instrument, and web design frameworks could all be covered. The suggestion of a method to automate program creation for user confluence simulation has contributed to the field of web development's Yang et al. [22] presented adds to the increasing body of static analysis work for Android software. Using a novel context-sensitive analysis of event handlers, an authority proceed description of end user-direct respond behaviour was built. Moreover, a user examination for building GUI models which has been included. Jurgela et al. [23] proposed MDA guidelines and UML models are used to create smart contract implementation code as well as to improve understanding and reuse of smart contracts. It will show UML Category and condition appliance illustration, the MDA principles, and modification algorithms it will be profitably hanged down to create consistency intelligent agreement for Ethereum network. S Yang et al. [24] proposed the cornerstone for fixed investigation for the Android as been representation for window call-back sequences. The inter process authority proceed the graph, crucial visual of conventional fixed investigation, can be used as an analogue. Provide algorithms for the development and traversal of the WTG have resemblance significant fixed representation for Android.. McQuistin et al. [25] presented It is possible to achieve this while keeping accuracy and performance by using apparatus it will be impulsively construct the prototype agreement interpret performance for the quality certificate which will describe it. We have demonstrated the ability to produce an attentive, right decipher execution it will process entering data in the cost of Gbit by second using a description of TCP in a style that is substantially similar to that in ordinary use. X Meng et al. [26] present edit is demonstrated that the WADE IDE

can be helpful for implementing a number of GUI-based changes in pre-existing software. While similar improvements are possible using alternative methods, the user study that is being reported here demonstrates that WADE greatly decreases the necessary knowledge and effort barriers. Cheng et al. [27] presented a multi-stage view point to the challenge of creating image captions that is based on visual-semantic attention. The version is built in hierarchical and up turned construction in series of adapter units, where all adapter unit will upgrade the concealed circumstances of all adapter unit connected to one at a time all duration stride. Q. You et al. [28] proposed a brand-new approach to annotating images that yields cutting-edge results on widely used benchmarks. Different Our method, which builds on earlier research, connect hierarchical and upturned techniques for excerpt deeper facts from picture. It then combines these techniques with an RNN that can focus specifically on rich exposition qualities found in picture. J. Ba et al. [29] proposed a novel computer vision model that utilises an attention mechanism to select the area of computation to concentrate on and demonstrated how it can be trained end-to-end to sequentially categorise many objects in an image. The model outperformed the most advanced Convnets on a task requiring recognition of multi-digit house numbers. X. Chen et al. [30] presented bidirectional RNN model's performance on several tasks and begin by outlining the training and testing datasets, then our baselines. The initial series of tests gauges how well our model can come up with original descriptions of pictures. The model's conducting in one and another determination repossession and picture retrieval tasks because it is bi-directional.

### III. DISCUSSION

Fig. 1 [20] Example of image mock-ups for code generation. First image mock-ups contain all the information to generate the code. This image mock-up shave been detected. After the detection cropping of the object as taken place which contain information for example it will detect and crop the button as written in the image mock-ups. Then recognition of object will take from that code has been generated. Table 1 [1] provides an illustration categorized GUI element representation data file, broken down by class. Which help to estimate CNN potency and it has been trained to prevent over fitting. Table 2 [1] illustrate the uncertainty pattern of control REDRAW and in Table 3 [1] illustrate uncertainty pattern of control Bag of Visual Words. Fig. 2 [1] A-C. illustrate the Grading correlation form on revise space. X and Y axis shows the edit distance and the fraction of penalty for the REMAUI Mock-ups hierarchies. It provides the ranking that near to the quarry ranking. The three class of functioning like insertion edits, deletion edits, substitution edits has been taken place to contest the quarry ranking.

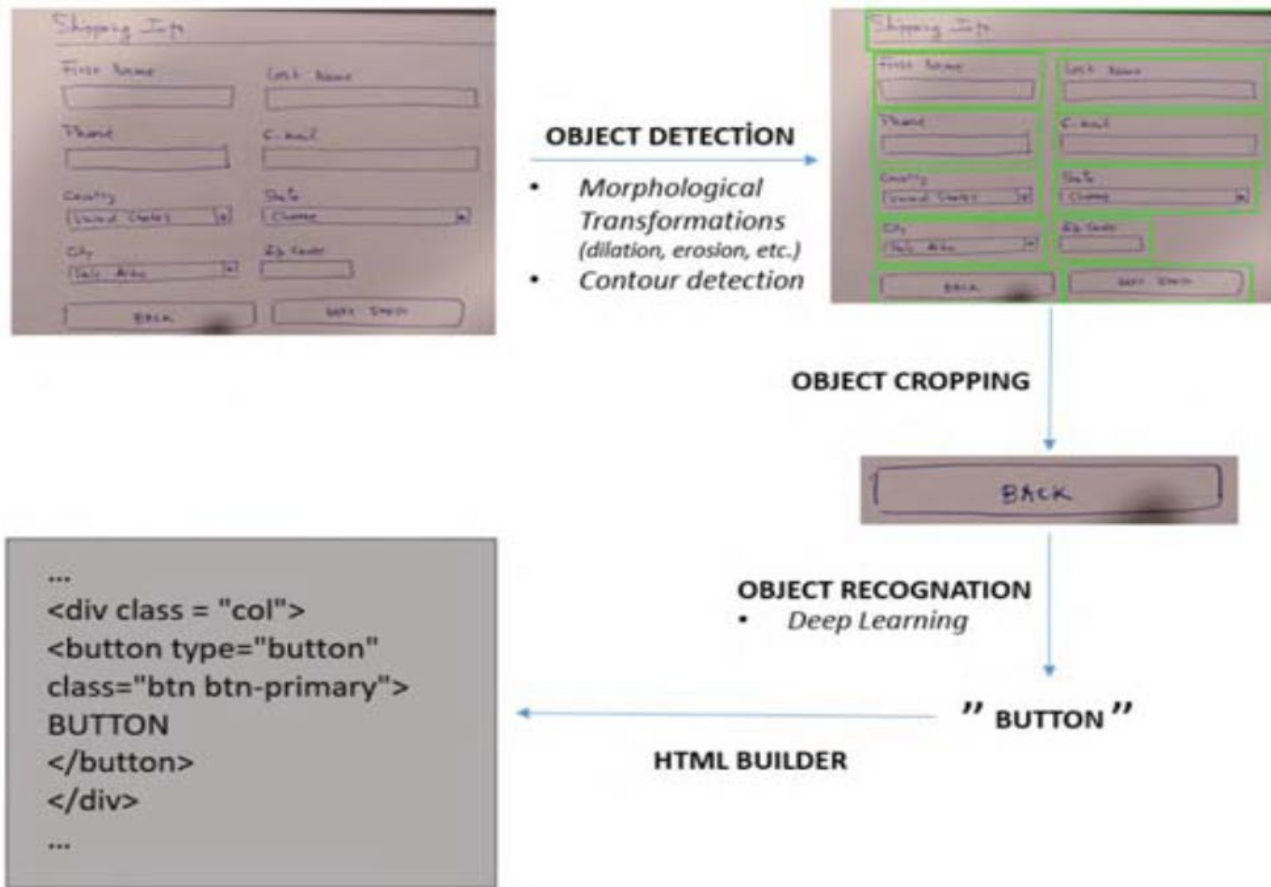


Fig. 1. Example of image mock-ups for code generation.

TABLE 1: Categorized GUI element representation data file

GUI-C Type	Total # (C)	Total (O)	Tr(O+S)	Valid	Test
TextView	99,200	74,087	74,087	15,236	9,877
ImageView	53,324	39,983	39,983	7,996	5,345
Button	16,007	12,007	12,007	2,400	1,600
ImageButton	8,693	6,521	6,521	1,306	866
EditText	5,643	4,230	5,000	846	567
CheckedTextView	3,424	2,582	5,000	505	337
CheckBox	1,650	1,238	5,000	247	165
RadioButton	1,293	970	5,000	194	129
ProgressBar	406	307	5,000	60	39
SeekBar	405	304	5,000	61	40
NumberPicker	378	283	5,000	57	38



Switch	373	280	5,000	56	37
ToggleButton	265	199	5,000	40	26
RatingBar	219	164	5,000	33	22
Spinner	20	15	5,000	3	2
Total	191,300	143,170	187,598	29,040	19,090

TABLE 2: Uncertainty pattern of control REDRAW

	Total	TV	IV	Bt	S	ET	IBt	CTV	PB	RB	TB	CB	Sp	SB	NP	RBt
TV	9877	94%	3%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
IV	5345	5%	93%	1%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Bt	1600	11%	6%	81%	0%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
S	37	5%	3%	0%	87%	0%	0%	5%	0%	0%	0%	0%	0%	0%	0%	0%
ET	567	14%	3%	2%	0%	81%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
IBt	866	4%	23%	1%	0%	0%	72%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CTV	337	7%	0%	0%	0%	0%	0%	93%	0%	0%	0%	0%	0%	0%	0%	0%
PB	41	15%	29%	0%	0%	0%	0%	0%	56%	0%	0%	0%	0%	0%	0%	0%
RB	22	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
TBt	26	19%	22%	7%	0%	0%	0%	0%	0%	0%	52%	0%	0%	0%	0%	0%
CB	165	12%	7%	0%	0%	1%	0%	0%	0%	0%	0%	81%	0%	0%	0%	0%
Sp	2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%
SB	39	10%	13%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	78%	0%	0%
NP	40	0%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	95%	0%
RBt	129	4%	3%	2%	0%	0%	0%	1%	0%	0%	0%	1%	0%	0%	0%	89%

TABLE 3: Uncertainty pattern of control Bag of Visual Words

	Total	TV	IV	Bt	S	ET	IBt	CTV	PB	RB	TB	CB	Sp	SB	NP	RBt
TV	9877	59%	4%	9%	1%	6%	2%	8%	6%	0%	1%	2%	0%	1%	0%	2%
IV	5345	4%	51%	4%	1%	2%	11%	2%	18%	1%	1%	3%	0%	2%	0%	2%
Bt	1600	6%	6%	59%	1%	5%	4%	7%	4%	0%	1%	1%	0%	0%	3%	1%
S	37	5%	0%	3%	65%	0%	0%	5%	22%	0%	0%	0%	0%	0%	0%	0%
ET	567	6%	2%	4%	1%	62%	1%	4%	15%	0%	0%	1%	0%	0%	4%	1%
IBt	866	2%	16%	3%	0%	2%	61%	1%	9%	1%	1%	2%	0%	2%	0%	3%
CTV	337	3%	1%	7%	1%	3%	0%	81%	1%	0%	0%	2%	0%	0%	0%	2%
PB	41	0%	24%	2%	0%	2%	5%	2%	54%	0%	0%	2%	2%	2%	0%	2%
RB	22	0%	5%	0%	0%	0%	0%	0%	27%	68%	0%	0%	0%	0%	0%	0%
TBt	26	7%	7%	19%	0%	0%	0%	11%	15%	0%	33%	0%	0%	0%	0%	7%
CB	165	4%	2%	3%	1%	2%	1%	2%	12%	1%	0%	72%	0%	0%	0%	1%
Sp	2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%
SB	39	0%	5%	0%	0%	0%	0%	0%	18%	3%	0%	5%	0%	68%	0%	3%
NP	40	3%	0%	5%	0%	3%	0%	5%	0%	0%	0%	0%	0%	0%	84%	0%
RBt	129	6%	3%	5%	1%	3%	0%	6%	18%	0%	0%	1%	0%	1%	0%	55%

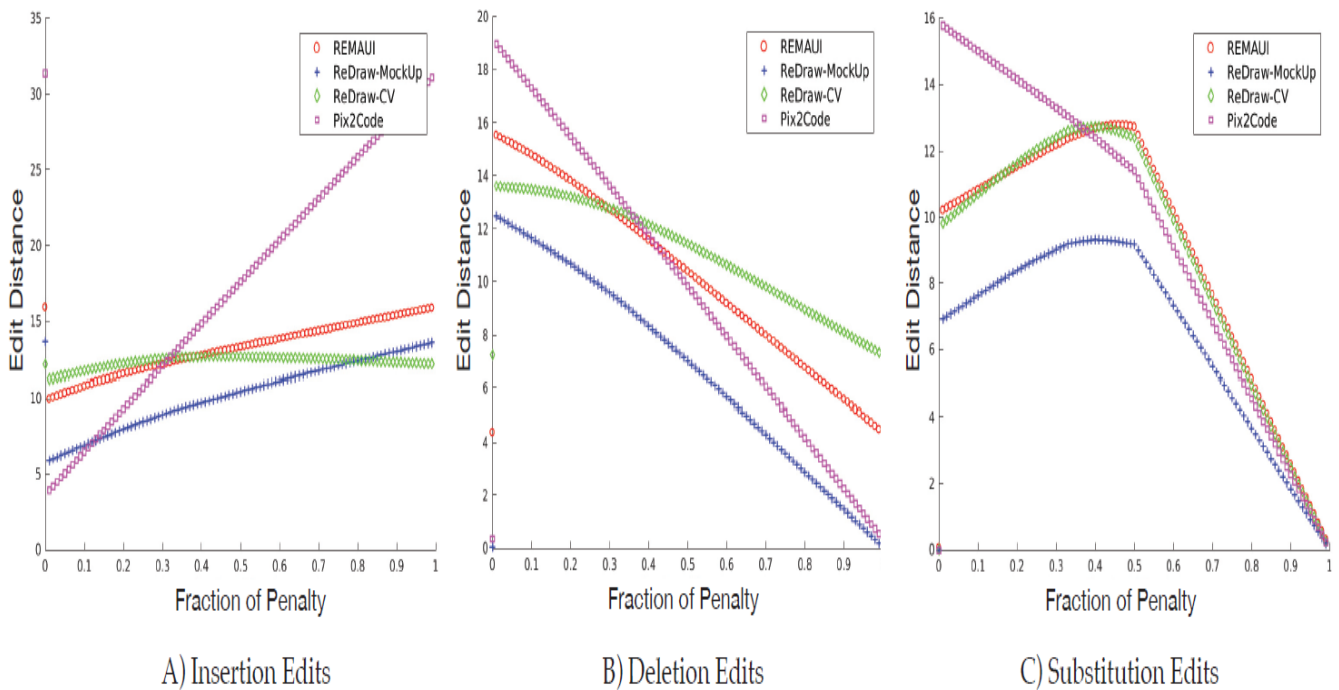


Fig. 2: Grading correlation form on revise space

#### IV. CONCLUSION

In this study, an automatic code generation is a process that will generate a program from code and data in a human readable way. The main cause of this paper is to represent an automatic code generation algorithm that uses image mock-ups as input for generating code. Such an algorithm could be used by developers who do not have expertise in computer graphics and need to quickly develop programs with high quality, or for anyone else who needs to create some application logic from scratch. The advantage of code generation from image mock-ups is that it provides a dynamic UI, which can be generated on demand. This can significantly reduce the time and cost associated with the development process. It provides a single source of truth for the entire interface or web page. The developer can then reuse this information in multiple places in their application and ensuring consistency.

#### V. REFERENCES

- [1]. K. P. Moran, C. Bernal-C'ardenas, M. Curcio, R. Bonett, and D. Poshyvanyk, "Machine learning-based prototyping of graphical user interfaces for mobile apps," *IEEE Transactions on Software Engineering*, pp. 1–1, 2018.
- [2]. T. A. Nguyen and C. Csallner, "Reverse Engineering Mobile Application User Interfaces with REMAUI (T)," in 2015 30<sup>th</sup> IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, nov 2015, pp. 248–259. [Online]. Available: <http://ieeexplore.ieee.org/document/7372013/>
- [3]. S. Natarajan and C. Csallner, "P2A: A Tool for Converting Pixelsto Animated Mobile Application User Interfaces," *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems - MOBILESoft '18*, pp. 224–235, 2018. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3197231.3197249>
- [4]. T. Beltramelli, "pix2code: Generating code from a graphical user interface screenshot," *CoRR*, vol. abs/1705.07962, 2017. [Online]. Available: <http://arxiv.org/abs/1705.07962>
- [5]. S. P. Reiss, Y. Miao, and Q. Xin, "Seeking the user interface," *Automated Software Engineering*, vol. 25, no. 1, pp. 157–193, mar 2018. [Online]. Available: <https://doi.org/10.1007/s10515-017-0216-3>
- [6]. A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [7]. S. Barnett, R. Vasa, and J. Grundy, "Bootstrapping mobile app development," in *Proc. 37th ACM/IEEE*



- International Conference on Software Engineering (ICSE). IEEE, May 2015.
- [8]. M. Gargenta and M. Nakamura, *Learning Android: Develop Mobile Apps Using Java and Eclipse*, 2nd ed. O'Reilly, Jan. 2014.
- [9]. M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow. *Deepcoder: Learning to write programs*. arXiv preprint arXiv:1611.01989, 2016.
- [10]. B. Dai, D. Lin, R. Urtasun, and S. Fidler. *Towards diverse and natural image descriptions via a conditional gan*. arXiv preprint arXiv:1703.06029, 2017.
- [11]. W. Ling, E. Grefenstette, K. M. Hermann, T. Kočiský, A. Senior, F. Wang, and P. Blunsom. *Latent predictor networks for code generation*. arXiv preprint arXiv:1603.06744, 2016.
- [12]. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. *Generative adversarial text to image synthesis*. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 3, 2016.
- [13]. H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. *Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks*. arXiv preprint arXiv:1612.03242, 2016.
- [14]. K. Moran, B. Li, C. Bernal-Cárdenas, D. Jelf, and D. Poshyvanyk, "Automated reporting of gui design violations in mobile apps," in *Proceedings of the 40th International Conference on Software Engineering Companion*, ser. ICSE '18. Piscataway, NJ, USA: IEEE Press, 2018, p. to appear.
- [15]. K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, and D. Poshyvanyk, "Automatically discovering, reporting and reproducing android application crashes," in *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation (ICST'16)*. IEEE, 2016, pp. 33–44.
- [16]. B. Deka, Z. Huang, C. Franzen, J. Hibschan, D. Afergan, Y. Li, J. Nichols, and R. Kumar, "Rico: A mobile app dataset for building data-driven design applications," in *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, ser. UIST '17, 2017.
- [17]. K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, and D. Poshyvanyk, "Crashscope: A practical tool for automated testing of android applications," in *Proceedings of the 39th International Conference on Software Engineering Companion*, ser. ICSE-C '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 15–18. [Online]. Available: <https://doi.org/10.1109/ICSE-C.2017.16>
- [18]. K. Mao, M. Harman, and Y. Jia, "Sapienz: Multi-objective automated testing for android applications," in *Proceedings of the 25th International Symposium on Software Testing and Analysis*, ser. ISSTA'16. New York, NY, USA: ACM, 2016, pp. 94–105. [Online]. Available: <http://doi.acm.org/10.1145/2931037.2931054>
- [19]. W. Fu and T. Menzies, "Easy over hard: A case study on deep learning," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE'17. New York, NY, USA: ACM, 2017, pp. 49–60. [Online]. Available: <http://doi.acm.org/10.1145/3106237.3106256>
- [20]. H. Dave, S. Sonje, J. Pardeshi, S. Chaudhari and P. Raundale, "A survey on Artificial Intelligence based techniques to convert User Interface design mock-ups to code," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 28-33, doi: 10.1109/ICAIS50930.2021.9395994.
- [21]. C. Machado and J. C. Campos, "Towards the integration of user interface prototyping and model-based development," 2021 International Conference on Graphics and Interaction (ICGI), Porto, Portugal, 2021, pp. 1-8, doi: 10.1109/ICGI54032.2021.9655284.
- [22]. Shengqian Yang, Dacong Yan, Haowei Wu, Yan Wang, and Atanas Rountev. 2015. *Static control-flow analysis of user-driven callbacks in Android applications*. In *Proc. 37th IEEE/ACM International Conference on Software Engineering (ICSE)*. IEEE, 89–99.
- [23]. M. Jurgelaitis, L. čepionienė and R. Butkienė, "Solidity Code Generation From UML State Machines in Model-Driven Smart Contract Development," in *IEEE Access*, vol. 10, pp. 33465-33481, 2022, doi: 10.1109/ACCESS.2022.3162227.
- [24]. Shengqian Yang, Hailong Zhang, Haowei Wu, Yan Wang, Dacong Yan, and Atanas Rountev. 2015. *Static window transition graphs for Android*. In *Proc. 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 658–668.
- [25]. S. McQuistin, V. Band, D. Jacob and C. Perkins, "Investigating Automatic Code Generation for Network Packet Parsing," 2021 IFIP Networking Conference (IFIP Networking), Espoo and Helsinki, Finland, 2021, pp. 1-9, doi: 10.23919/IFIPNetworking52078.2021.9472829.
- [26]. Xiaojun Meng, Shengdong Zhao, Yongfeng Huang, Zhongyuan Zhang, James Eagan, and Ramanathan Subramanian. 2014. *WADE: simplified GUI add-on development for third-party software*. In *Proc.*



- ACM SIGCHI Conference on Human Factors in Computing Systems (CHI). ACM, 2221–2230.
- [27]. L. Cheng, W. Wei, X. Mao, Y. Liu and C. Miao, "Stack-VS: Stacked Visual-Semantic Attention for Image Caption Generation," in *IEEE Access*, vol. 8, pp. 154953-154965, 2020, doi: 10.1109/ACCESS.2020.3018752.
- [28]. Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.
- [29]. J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *ICLR*, 2015.
- [30]. X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, pages 2422–2431, 2015.