

An Automatic Contrast Validation Approach for Smartphone Themes

Aasim khurshid, Adriano Gil, Felipe Augusto Souza guimarães, Mikhail Gadelha , Everlandio Fernandes
Sidia Instituto de Ciência e Tecnologia

AM, Brazil

Email: aasim.khurshid,adriano.gil,felipe.guimaraes,mikhail.gadelha,everlandio.fernandes@sidia.com

Abstract—The smartphone themes submitted to the theme-store are first evaluated by human experts to ensure a pleasant visual experience¹. One of the main challenges in smartphone themes evaluation is to validate the contrast of the elements of the theme. Contrast refers to the difference in visual properties that makes an object distinguishable from other objects and the background. In this work, we propose an automatic themes evaluation approach that validates the contrast of Android smartphone themes among regular and non-regular at the element level. To localize the contrast affected regions, the proposed themes evaluation is divided into two phases: 1) Element extraction; and 2) Contrast evaluation. For element extraction, a customized framework is created that utilizes native Android frameworks. Following, these elements are analyzed by using a specifically designed Convolutional Neural Network (CNN) for smartphone theme elements contrast evaluation. The proposed approach is evaluated on two databases which are composed of using the real Android themes. Experiments suggest that our proposed contrast-detection network can obtain a better performance than the known state-of-the-art classification methods, in smartphone themes evaluation, based on evaluation measures like Accuracy, F1 score, processing time and others.

I. INTRODUCTION

Themes are an essential components of smartphones to provide a pleasant user experience. A theme is a visual style for a set of screen composed of multiple elements such as User Interface (UI) elements, backgrounds, icon packs, and other elements. Some example elements can be seen in the middle of Figure 1. Moreover, themes usually contain colorful background, artistic and textual features. One of the main challenges in the themes evaluation process is to validate the contrast between themes elements. Contrast refers to the difference in visual properties that makes an object distinguishable from other objects and the background. The contrast is described as High Perceivable Contrast (HPC) if all the elements are distinguishable from other elements and background, otherwise, it is referred to as a Low Perceivable Contrast (LPC). These themes are usually evaluated for contrast by the theme validation experts to qualify to sell on the theme store [1]. As the human visual system is more sensitive to contrast than absolute luminance; therefore, humans can perceive the world similarly regardless of the considerable changes in illumination conditions, this can be justified. However, this is a time-consuming and expensive process, especially when there

is substantial progress in image processing technologies and these technologies can provide a comprehensive, faster, and cheaper solution for contrast validation in smartphone themes.

In themes, it is required that all the elements have high perceivable contrast with the background and text in that element. This requires accurate detection of the elements and the hierarchy of the screenshot plays an important role. Themes elements and hierarchy detection are explored by researchers recently, such as Reverse Engineer Mobile Application User Interfaces (REMAUI) uses edge detection, dilation and Optical Character Recognition (OCR) techniques [2] to identify UI elements such as images, texts, containers, and lists. Similarly, Mozgovoy et al. utilized pattern matching techniques to test hand-drawn user interfaces [3].

Contrast enhancement is a well-studied field in image processing and huge research on improvement of the contrast in images. These techniques include histogram-based approaches [4], [5], morphological operations for image enhancements [6], transform-based methods such as Discrete Cosine Transform (DCT) [7], optimization-based contrast enhancement approach using Genetic Algorithm (CEBGA) to maximize a contrast measure based on edge information [8] and CNN for contrast enhancement [9]. To measure the contrast, these methods usually use Peak Signal to Noise Ratio (PSNR), Tenengrad measure [10], or the difference between the darkest and the brightest pixel in an image-contrast ratio. PSNR method requires a reference image to compare with, tenengrad measure computes the variance of Laplacian and works well for focus detection, whereas contrast ratio is not an effective method because it has a very limited understanding of the image [10]. Furthermore, global contrast measure based on histograms is used for salient object detection, however, it does not align directly with themes evaluation which requires local contrast [11]. Also, contrast detection is explored using human experts [12] and hardware-based pixel arrays [13], but these methods are impractical and expensive for themes evaluation due to a large number of themes and screen levels. These methods explained above do not fit directly to the contrast detection for smartphone themes evaluation, however, these methods may be extended for contrast detection instead of an enhancement.

Moreover, in themes, it is not only required to detect contrast, but also the exact location of the low perceivable contrast (LPC) elements to help designers correct it easily, because

¹<https://developer.samsung.com/sdp/blog/en-us/2020/02/07/how-to-become-a-samsung-mobile-ui-themes-designer>

of many screens in a smartphone theme. Fernandes et al. proposed a themes evaluation tool based on object detection and contrast detection [14]. However, the accuracy of the theme evaluation is highly dependent on object detection accuracy. The proposed themes evaluation tool solves this problem by using a customized framework to extract elements. The proposed approach has two phases, which are: 1) Element extraction, and 2) Contrast evaluation. The proposed automatic themes evaluation scheme validates the contrast of android smartphone themes² among regular and non-regular at the element level.

The elements are extracted using a customized framework that utilizes two native android frameworks: the UI Automator [15] and the Accessibility service [16]. The UI Automator is used to simulate user inputs to reach the screens affected by themes and the Accessibility service is used to extract the XML, which allows extracting all the screen elements. The strategy simplifies our scheme by avoiding explicit object detection as proposed by Fernandes et al [14].

For classification, these theme elements are divided into regular and non-regular theme elements. The regular theme element means that this particular element has a High Perceivable Contrast (HPC) and irregular refers to the element which has Low Perceivable Contrast (LPC).

These elements are analyzed by using a contrast detection CNN, that evaluates the contrast of each element of the theme independently. As the elements are usually small and contain limited information, it is more interesting to create a small network architecture that can classify among elements containing or not containing contrast issues. Furthermore, small networks are fast and are less prone to over-fitting.

The rest of the paper is organized as follows: Section II provides a detailed explanation of the proposed approach. Next, experimental results and discussion are presented in section II, and lastly, the paper provides conclusions and future work in section IV.

II. PROPOSED APPROACH

Figure 1 shows the working of the proposed approach for themes evaluation. Its two components: Element extraction, and Contrast evaluation works in sequence to perform contrast validation for the submitted Android theme. When a theme is submitted for evaluation, the element extraction module extracts all the elements and its location in the theme. The theme elements are passed to the contrast evaluation CNN, which detects if the element tested has HPC or LPC. After the contrast evaluation, these elements detection result is reported and displayed on the theme screen by putting red bounding boxes on the elements affected by contrast issues (LPC elements), and green bounding boxes for the elements that do not have any contrast issue (HPC elements).

A. Element Extraction

The elements are extracted using a customized framework that utilizes two native Android frameworks: the UI Automator [15] and the Accessibility Service [16]. The UI Automator

is used to simulate user inputs to reach the screens affected by changing the device theme, and the Accessibility Service is used to retrieve the UI hierarchy of the current screen: it is an XML file with a tree of UI elements and their properties, e.g., if the UI element contains a text if it has focus, etc. More importantly, the XML contains the bounding box of each UI element on the screen, i.e., its position on the screen, its height and its width. By using the bounding box of each UI element our scheme only has to evaluate the element contrast, instead of detecting the UI element and evaluating contrast in the same network as proposed by Fernandes et al [14].

B. Contrast Evaluation

For classification, these theme elements are divided into regular and non-regular theme elements. The regular theme element means that this particular element has a High Perceivable Contrast (HPC) and irregular refers to the element which has Low Perceivable Contrast (LPC).

These elements are analyzed by using a contrast detection Convolutional Neural Network (CNN). As the elements are usually small and contain limited information, it is more interesting to create a network architecture with a small number of layers that can classify among elements containing or not containing contrast issues. Furthermore, small networks are fast and are less prone to overfitting. The network architecture is explained next.

a) Network Architecture: The architecture of the proposed contrast-detection network is composed of 5 sets of layers to learn hyper-parameters to classify element contrast into HPC (regular) and LPC (non-regular). The contrast evaluation at the element level simplifies the problem and the model learn a small number of hyper-parameters, which during the testing phase, achieves higher speed and has low computational complexity.

Figure 2 shows the proposed network architecture. Apart from the input and output layers, each layer of the network is composed of multiple two-dimensional planes, and each plane contains an independent neural unit. The proposed network connects the convolutional layer, batch normalization layer, Rectified Linear Units(ReLU) Layer, and pooling layer successively from left to right. The batch normalization layer is used to stabilize the learning process and reduce the number of training epochs required to train the small-CNN. Furthermore, the ReLU layer introduces non-linearity and reduces the vanishing gradient problem, which may slow the convergence of the last layers, otherwise. This layer changes all the negative activations to 0. Finally, the max-pooling layer is used to reduce the spatial dimensions, which helps decrease the number of hyper-parameters to reduce computational cost and also avoids over-fitting of data. After, these layers, a fully connected layer with two classes (regular and non-regular) is used, followed by a softmax Layer and the classification Layer to classify different scenes.

III. EXPERIMENTAL EVALUATION AND DISCUSSION

The proposed themes evaluation scheme is trained and tested on two datasets of themes elements: Darkset, which contains

²https://www.samsung.com/hk_en/apps/galaxy-themes/

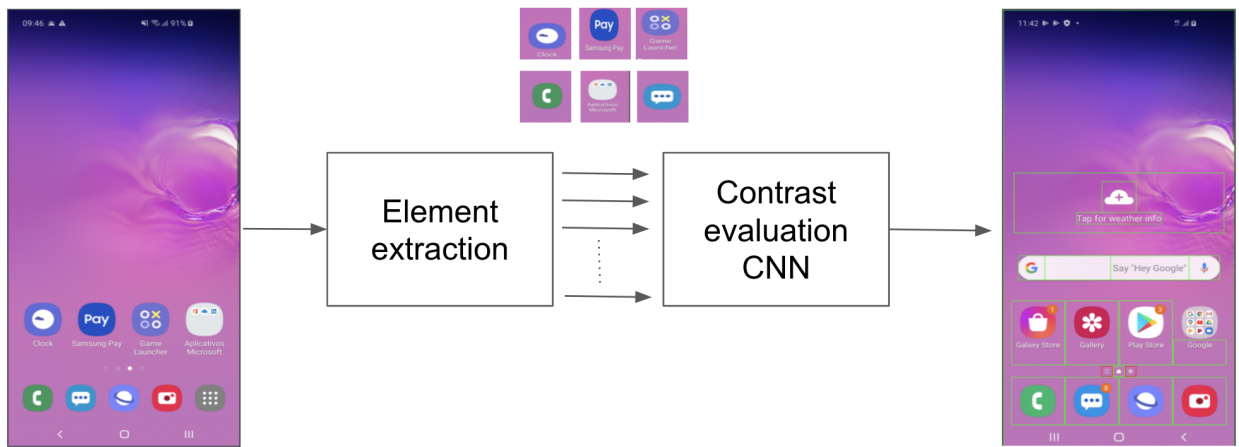


Fig. 1. Themes evaluation approach.

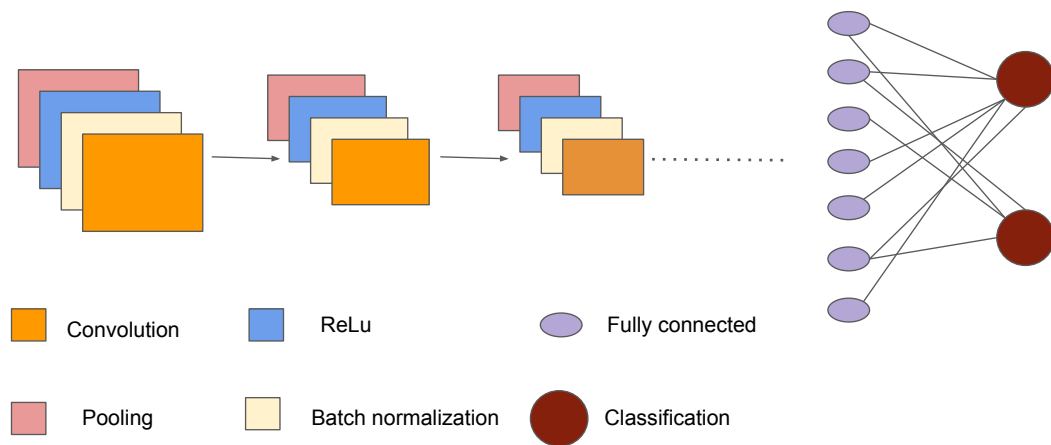


Fig. 2. Convolutional Neural Network architecture for theme element contrast detection.

themes predominately dark color themes; and Lightset, which contains predominantly light color themes. Moreover, these theme elements extracted from real smartphone themes, where each theme contains multiple screens including the home screen and application-specific screens. More than 50 themes were used to create the datasets and each theme contains 122 screens across a variety of applications (for instance, settings, calendar, my files). Each screen has one to ten elements, which may refer to an application icon, text, and background as can be seen in Figures 3 and 4. For evaluation, the dataset is divided into training, validation, and a test set of regular (HPC) and non-regular (LPC) theme elements. Table I shows details of the training set, validation set, and test set of these two datasets. The Darkset contains 29390 training elements, 14658 validation, and 4520 test elements. Moreover, the Lightset contains 26680 theme elements for training, 13360 for validation, and 4296 for the test, respectively.

Moreover, the experiments are executed on a MacBook Pro 2018 using a 2,6GHz 6-Core Intel Core i7 and GPU Radeon

Dataset	Dark	Light	Total
Training	29390	26680	56070
Validation	14658	13360	28018
Test	4520	4296	8816

TABLE I
DATASET PARTITIONING INTO TRAINING, VALIDATION AND TEST SET.

Pro 560X 4GB. Furthermore, PlaidML³ was used as Deep-learning backend that allows to use GPU for generic purposes in MacOS platforms.

Figure 3 shows an example of the default screen and the analyzed theme when the theme is applied to the default screen using the proposed theme evaluation scheme. The green bounding boxes show the elements with HPC and red bounding box show LPC components detected by using the proposed theme evaluation approach. For example, in Figure 3(b) the icon for "cancel" and "save" have LPC, while "Gallery" and "camera" have HPC. Figure 4 shows some example of analyzed

³<https://readthedocs.com/projects/vertexai-plaidml/downloads/pdf/stable/>

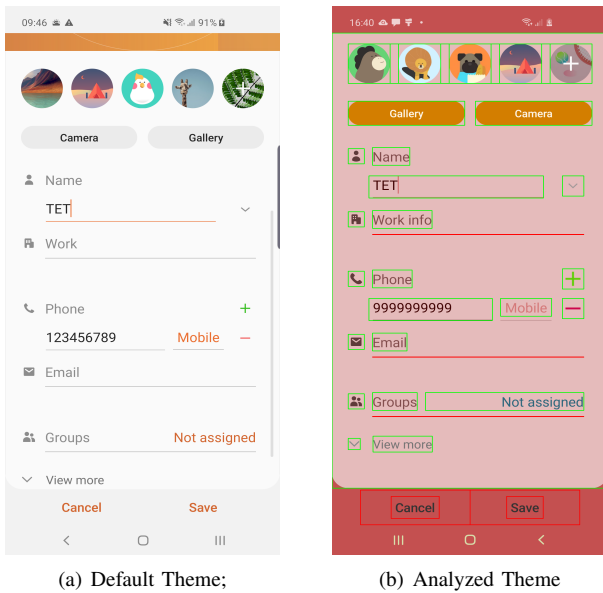


Fig. 3. Analyzed Theme.

themes, which shows the element extracted and tested using the proposed theme evaluation scheme. It can be seen that the proposed scheme extract elements precisely. The green bounding boxes show elements of HPC and the LPC bounding boxes are indicated in red color. Moreover, the proposed scheme is capable of detecting contrast issues in the evaluated themes accurately and can be used as an automatic theme evaluation scheme for highlighting contrast issues.

The proposed contrast evaluation network is compared quantitatively with the ResNet50 [17], AlexNet [18] and GoogleNet [19] representative of the state-of-the-art for classification, using the measures which include:

- Attack Presentation Classification Error Rate (APCER);
- Normal Presentation Classification Error Rate (NPCER);
- Average Classification Error Rate (ACER);
- False Positive Rate (FPR);
- True Positive Rate (TPR);
- True Negative Rate (TNR);
- Accuracy;
- F1 score and
- processing time

for contrast validation phase. It is to note that the comparative networks are fine-tuned for the contrast validation dataset using transfer learning approach by allowing to relearn the weights at every layer for improved performance [20]. Moreover, processing time for the whole dataset is considered because it depicts processing time in relation to the theme elements rather than the whole screen (frame), which may contain a varied number of elements and hence does not give a consistent processing time perspective. APCER, NPCER, and ACER are

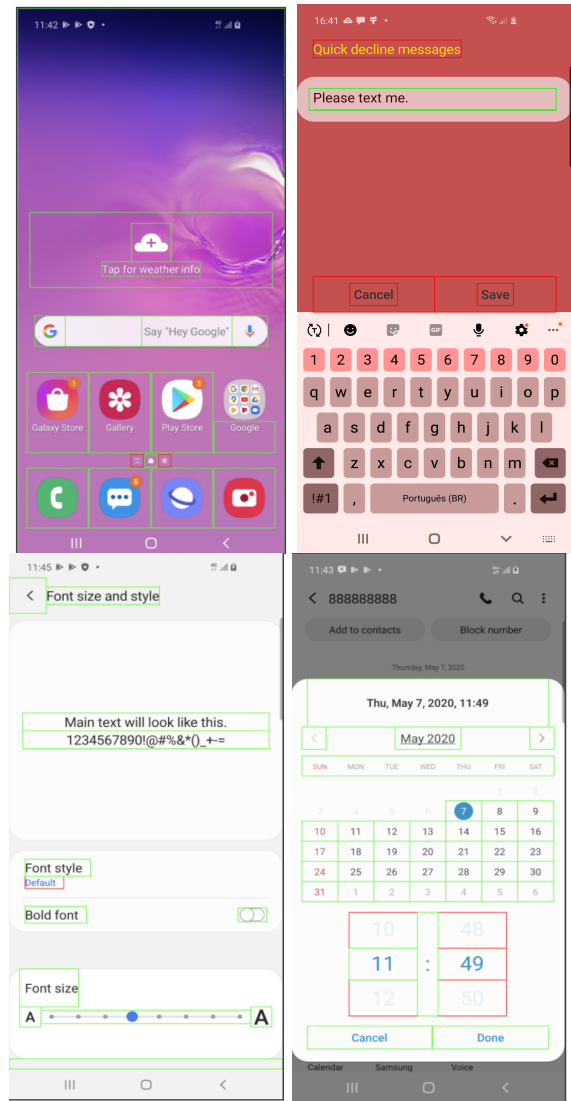


Fig. 4. Example of Analyzed Themes (left to right), top to bottom; home screen; Contact; Font size; Calendar.

the classification error measures and can be computed as [21]:

$$\begin{aligned}
 APCER &= \frac{FP}{TN + FP} \\
 NPCER &= \frac{FN}{FN + TP} \\
 ACER &= \frac{APCER + NPCER}{2}
 \end{aligned} \tag{1}$$

It is to note that, for all the comparative methods, the elements are extracted using the proposed element extraction method, and the same training, evaluation, and test data is used to make a fair comparison.

Table II provide the quantitative results of the proposed contrast detection and the comparative method. It can be seen from the table that the proposed method outperforms the deeper networks, fine-tuned for contrast validation in themes, based on most of the metrics used, and confers a small average classification error rate on both the datasets, which

Method	APCER	NPCER	ACER	FPR	TPR	TNR	Accuracy	F1 Score	Processing time (seconds)
Training and test data extracted from Lightset dataset									
ResNet [17]	0.1821	0.1123	0.1507	0.2150	0.9032	0.7937	0.8445	0.8543	269.01
AlexNet [18]	0.0048	0.9956	0.5002	0.0048	0.9956	0.5002	0.4838	0.3175	184.79
GoogleNet [19]	0.0241	0.0195	0.0218	0.0241	0.9805	0.9759	0.9781	0.9772	160.08
Proposed CNN	0.0517	0.0239	0.0378	0.0517	0.9761	0.9483	0.9616	0.9605	56.16
Training and test data extracted from Darkset dataset									
ResNet [17]	0.2801	0.0731	0.1767	0.3721	0.9541	0.6383	0.7895	0.7915	313.24
AlexNet [18]	0.0625	0.088	0.0813	0.0634	0.9455	0.9032	0.9244	0.9276	183.22
GoogleNet [19]	0.1855	0.0302	0.1078	0.1855	0.9698	0.8145	0.8887	0.8928	174.21
Proposed CNN	0.0817	0.0444	0.0631	0.0817	0.9556	0.9183	0.9354	0.9314	49.08

TABLE II
EXPERIMENTAL EVALUATION OF THE PROPOSED APPROACH AND COMPARATIVE METHODS (BEST RESULTS IN **BOLD**); SECOND BEST IN *bold-italic*.

is important in the case of themes evaluation to minimize the unnecessary returning the theme to the designer. ResNet50 [17] performs badly on both the datasets used and does not provide satisfactory performance in the themes elements contrast evaluation. On the other hand, fine-tuned AlexNet [18] has the smallest FPR on both the datasets, but it performs differently for the different datasets. For example, it has the smallest TNR and consequently reduces accuracy and F1 score to 0.48 and 0.31 on the lightset database, which is undesirable. This behavior makes its use unsuitable for contrast evaluation. Moreover, GoogleNet [19] performs fairly well on both the dataset, however, it also has a difference of 0.8 points among two different datasets. The proposed method has performed consistently well for both the datasets and only has a difference of 0.3 points across datasets. Furthermore, the proposed method outperforms all the other methods on the Darkset and is second on the lightset in terms of accuracy and F1 score to GoogleNet [19] considering that the proposed CNN has just 5 layers that take 253 minutes to train compared to the 22 layer GoogleNet that requires 681 minutes fine-tune on the Darkset, for reference.

In terms of processing time, the proposed method is at least three times faster than all the comparative methods. The proposed method evaluates the contrast of all the test elements of the lightset in 56.16 seconds compared to 160.08 seconds of the second-fastest comparative method, which is a huge improvement. We can also see a similar trend for the Darknet dataset. This efficiency is attributed to its small architecture, which in turn learns a small number of efficient parameters for elements contrast detection in smartphone themes.

Finally, it is clear from Table II that the proposed method performs consistently better than the comparative CNNs based on average classification error rate (ACER), TNR, accuracy, F1 score and time on both the darkset, and stands second in terms of ACER, accuracy and F1 score on the lightset with a margin of just 0.1 points but still stands out in terms of its

processing time. These measures prove the effectiveness of the proposed small CNN in the context of themes element contrast detection for smartphone theme evaluation.

IV. CONCLUSION

In this paper, we proposed a contrast validation scheme for Android smartphone themes evaluation. It is composed two components; element extraction and contrast evaluation. These two components work in sequence to evaluate each element of the applied theme and indicate the elements that have low perceivable contrast. Elements are extracted using a customized framework based on the native Android framework, which helps simplify the task of the contrast evaluation network. For elements contrast evaluation, a Convolutional Neural Network with a limited number of layers is composed for increased performance. The proposed contrast evaluation scheme performs better than the comparative state-of-the-art classification methods including deep CNNs, which proves the effectiveness of the proposed method. In the future, we look to improve further the accuracy while maintaining the time constraint.

ACKNOWLEDGMENT

This work is partially supported by Sidia Institute of Science and Technology, and Samsung Eletrônica da Amazônia Ltda, under the auspice of the Brazilian informatics law no 8.387/91.

REFERENCES

- [1] T. Morelan. (2020) How to become a samsung mobile ui themes designer. [Online]. Available: <https://developer.samsung.com/sdp/blog/en-us/2020/02/07/how-to-become-a-samsung-mobile-ui-themes-designer>
- [2] T. A. Nguyen and C. Csallner, "Reverse engineering mobile application user interfaces with remait," in *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2015, pp. 248–259.
- [3] M. Mozgovoy and E. Pyshkin, "Using image recognition for testing hand-drawn graphic user interfaces," in *11th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2017)*, 2017, pp. 12–16.

- [4] M. Abdullah-Al-Wadud, M. H. Kabir, M. A. A. Dewan, and O. Chae, "A dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, 2007.
- [5] T. Celik and T. Tjahjadi, "Automatic image equalization and contrast enhancement using gaussian mixture modeling," *IEEE transactions on image processing*, vol. 21, no. 1, pp. 145–156, 2011.
- [6] A. Zadorozny and H. Zhang, "Contrast enhancement using morphological scale space," in *2009 IEEE International Conference on Automation and Logistics*. IEEE, 2009, pp. 804–807.
- [7] J. Tang, E. Peli, and S. Acton, "Image enhancement using a contrast measure in the compressed domain," *IEEE signal processing LETTERS*, vol. 10, no. 10, pp. 289–292, 2003.
- [8] S. Hashemi, S. Kiani, N. Noroozi, and M. E. Moghaddam, "An image contrast enhancement method based on genetic algorithm," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1816–1824, 2010.
- [9] Y. S. Moon, B. G. Han, H. S. Yang, and H. G. Lee, "Low Contrast Image Enhancement Using Convolutional Neural Network with Simple Reflection Model," *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 1, pp. 159–164, 2019.
- [10] M. Shah, S. Mishra, M. Sarkar, and C. Rout, "Identification of robust focus measure functions for the automated capturing of focused images from ziehl–neelsen stained sputum smear microscopy slide," *Cytometry Part A*, vol. 91, no. 8, pp. 800–809, 2017.
- [11] M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.
- [12] Q. Yu, P. Zhang, J. Qiu, and F. Fang, "Perceptual learning of contrast detection in the human lateral geniculate nucleus," *Current Biology*, vol. 26, no. 23, pp. 3176–3182, 2016.
- [13] Y. M. Wang, "Performing spatial and temporal image contrast detection in pixel array," Jan. 26 2016, uS Patent 9,247,109.
- [14] E. Fernandes, R. Correia, A. Gil, J. Postal, and M. R. Gadelha, "Themes validation tool," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 16–22.
- [15] N. Patil, D. Bhole, and P. Shete, "Enhanced ui automator viewer with improved android accessibility evaluation features," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. IEEE, 2016, pp. 977–983.
- [16] J. Kraunelis, Y. Chen, Z. Ling, X. Fu, and W. Zhao, "On malware leveraging the android accessibility framework," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 2013, pp. 512–523.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [19] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [20] A. Khurshid, S. C. Tamayo, and R. Grunitzki, "A scene classification approach for augmented reality devices," in *International Conference on Human-Computer Interaction*, vol. 46. Springer, 2020, p. LNCS 12428. [Online]. Available: https://www.researchgate.net/publication/343654512_A_Scene_Classification_Approach_for_Augmented_Reality_Devices
- [21] A. Khurshid, S. C. Tamayo, E. Fernandes, M. R. Gadelha, and M. Teofilo, "A robust and real-time face anti-spoofing method based on texture feature analysis," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 484–496.