Improving Optical Character Recognition on Partially Broken Khmer Characters in Printed Documents

Menghang Hean[1], Khem Raksa Peou[1], Neil Ian Cadungog-Uy[1], Hongly Va[2], Sa Math[3*], and Tharoeun Thap[3,*]

[1] Dept. of Computer Science, Paragon International University, Phnom Penh, Cambodia
[2] Cambodia Academy of Digital Technology, Phnom Penh, Cambodia
[3] Ministry of Post and Telecommunications, Phnom Penh, Cambodia
mhean@paragoniu.edu.kh; kpeou@paragoniu.edu.kh; nuy@paragoniu.edu.kh; Hongly.va@cadt.edu.kh; sa-math@mptc.gov.kh;thareoun-thap@mptc.gov.kh

Abstract. This study presents the solution to recognizing partially broken Khmer characters in printed text documents using image processing techniques and Machine Learning, specifically Deep Learning. It employs quantitative research along with the experimental design as it explores the relationship between different factors and the desired outcome by analyzing empirical data obtained from experimentation with different deep learning architectures. Otsu' Thresholding method and a morphological operation called Dilation are employed in this research as image processing techniques before passing the output into the optical character recognition process. The study performs a comparative analysis between two Deep Learning architectures, namely ResNet and VGG as feature extraction, in terms of speed and accuracy. On top of that, BiLSTM is used as a sequence modeling technique to perform contextual analysis coming from the feature extraction output before decoding into a series of words to form a sentence. The models are trained on a huge dataset of synthesized text images along with multiple Khmer fonts, where the images have lost partial visual representation to mimic the partially broken text that are found in historical documents. The two proposed models that are a combination of BiLSTM and CNN architecture (ResNet, VGG) outperformed Tesseract in terms of character error rate by 22% for the former and 18% for the latter.

Keywords: Optical Character Recognition, Partially Broken Khmer Characters, Khmer Text, BiLSTM.

1 Introduction

Artificial Intelligence (AI) is a subfield of Computer Science dedicated to the development of computer systems capable of executing tasks typically performed by humans,

* Co-corresponding Author: Sa Math and Tharoeun Thap
Acknowledgement: This research was supported by The Ministry of Post and Telecommunications, Kingdom of Cambodia.

including speech recognition, decision-making, and perceptual tasks. A particularly significant application of AI is Optical Character Recognition (OCR), which involves the conversion of images containing text—whether handwritten, typed, or printed—into machine-readable formats. This capability enables computers to extract and interpret textual information in a manner similar to human reading.

OCR technology is employed globally across a diverse range of applications. Key implementations include Vehicle Number Plate Recognition, utilizing OCR to identify and process vehicle registration plates for law enforcement and traffic management. Traffic Sign Recognition, enhancing autonomous vehicle navigation systems by recognizing and interpreting road signs. Document Processing, streamlining the extraction of information from scanned documents, such as checks, passports, invoices, bank statements, and receipts, which would otherwise necessitate manual data entry.

These applications underscore the transformative potential of OCR in automating data extraction processes, thereby increasing efficiency and accuracy in various sectors [1].

2 Literature Review

Limited research has been conducted thus far regarding the recognition of fragmented Khmer characters; however, valuable inspirations and insights can be derived from similar studies within the field.

Two studies implemented the usage of Image Processing techniques which serve as areas of exploration for the research. Praseetha and Deepa [2] mentioned that the reason that led to the failure of OCR recognition is the faulty character segmentation which mostly stemmed from the presence of broken characters. The study proposed an image processing technique known as the Active Contour Model, which refers to defining the boundary of objects to produce a generative and parametric curve or contour. This process aimed to improve the performance of character segmentation on Malayalam documents. Following this, feature extraction was performed on the extracted contours so that relevant data could be used for recognition. If the recognition is successful, the process is stopped. Otherwise, it combines the adjacent contours and goes through the recognition again. The proposed method is proven to be effective when the breakage width is not so wide.

Ramalingam and Bhojan [3] presented an approach to tackle the broken English characters issue by combining two popular image processing algorithms. The method proposed uses an image as input and applies a preprocessing technique called Otsu to convert the image into a noise-free binary format. Character segmentation is performed on the binary image, and the algorithms will detect whether the character is broken. The algorithm mainly classifies broken characters by checking whether the characters width exceeds 95% of the mean threshold. The space coming from broken characters that is less than the normal character space threshold will be merged together to form a whole character. The proposed method achieved an accuracy of 92.88% on English characters using Mean Based Thresholding combined with chain code algorithms. One study employed a neural network to restore the broken characters. Sandhya and Krishnan [4] proposed a new approach to addressing the broken characters of Kannada language

script using a neural network. In the training phase, the study conducted a process consisting of 7 steps, ranging from image preprocessing, segmentation, normalization, thinning, rebuilding, extraction of zonal features and neural network training. The process of rebuilding is performed by using the closing point algorithms to fill out the gaps found in broken characters based on zonal features. The neural network consists of three layers, which are the input layer, the hidden layer, and the output layer. The input layer in this study has 50 neurons, while the output layer consists of 49 neurons, denoting the number of character classes in Kannada. The study was reported to achieve 98.9% on the artificially generated dataset.

Three studies focused on deep learning approaches which provide the researchers with great insights and possibilities. Buoy et al. [5] from Techo Startup Cambodia proposed an end-to-end deep convolutional neural network that uses a sequence-to-sequence (Seq2Seq) architecture with an attention mechanism for Khmer OCR. The proposed method differs from the traditional method that utilizes a deep neural network with CTC in that it consists of two networks known as the encoder and decoder. The encoder network consists of two major components, which are convolutional blocks and recurrent layers, whereas the decoder network is made of a layer of GRU units and a linear classifier, utilizing the attention mechanism to process the encoder's output to predict the characters.

ResNet and VGG are two influential convolutional neural network architectures in the field of computer vision. ResNet introduced residual connections, enabling the training of deep networks by addressing the vanishing gradient problem. VGG, on the other hand, focused on increasing network depth with smaller convolutional filters to capture intricate visual patterns. Both architectures have made significant contributions to image recognition tasks [6, 7].

3 The Proposed Algorithm

3.1 Dataset Generation

For this research, data for training and testing are artificially generated as digital images with a white background using the Khmer word dataset from the GitHub repository mentioned above. The procedure begins with collecting different types of Khmer words and sentences and storing them in a file. Specific fonts are also collected to help generate the Khmer word correctly along with the background image that resembles the printed document in the real world. After completing the steps, a Python library-based data generator places text in the images to generate the desired amount specified by the researchers.

In order to generate Khmer partially broken characters, the researchers employ a series of steps to perform after generating the Khmer image text. The operation begins with generating random numbers across the images' dimensions, where the numbers are in the form of coordinates, in which the x-axis is in the range of the left 10% margin to the right 10% margin of the image's width. The same concept is applied to the y-axis of the number, but on images' height. After the generation of one coordinate, the process continues from that point and expands the range from both vertically and

horizontally followed by black-pixel replacement with white pixels to ensure that the text partially loses some of its visual representation. The operation is performed for a certain number of iterations based on the images' dimensions. Through trial-and-errors for image's dimension that is in the range from 32 × 100 to 32 × 300, the numbers of iterations vary from image to image based on its width. The image' width is divided into two groups: 100 - 150 pixels and 150 - 300 pixels, and the white-pixel masking operation will be applied to the former for 20 iterations and 40 on the latter. The number of iterations needs to vary based on the width as extremely low or high values of this number could potentially compromise image quality. In this research, 1 million text images are generated for the dataset and they are split into the following partitions:

- Training (85 %)
- Validation (5%)
- Testing (10 %)

3.2 Image Processing

In this research, the researchers decided to choose an image processing technique under the branch of Morphological Operations, particularly dilation, as a technique to connect the partially broken characters together. The algorithm begins with performing binarization; in this study, the researcher selected Otsu's Method [2] as one of the techniques to binarize. Following this, a structuring element, which is a matrix of pixel values, is chosen so that the process of dilation understands how many pixels need to be expanded We apply the dilation operation to the image, to fill up the holes based on their neighboring pixels, inspired by previous study in [10].

Fig. 1. Dataset Sample Before and After Processing

3.3 Feature Extraction

Feature Extraction plays a significant role in text recognition for its ability to capture the hierarchical features from the images in many levels. This method has been made popular and proven to be effective, with the evolution of Convolutional Neural Network (CNN). In this paper, the researchers employ two popular CNN architectures, namely ResNet and VGG for feature extraction. In Table 1 and 2, 'F', 'K', 'S' stands for Filter Size, Kernel, and Stride respectively.

Residual Neural Network (ResNet).

Residual Neural Network (ResNet) is a deep learning architecture that has been renowned for its ability to largely address the vanishing gradient problem, which helps the model to converge. The innate architecture of having residual blocks is what makes ResNet differ from traditional architectures. The residual block in ResNet is designed

to enable the network to learn and propagate residual information through the layers. This helps address the vanishing gradient problem and enables the training of very deep neural networks. The residual connections in the block allow the gradients to flow directly from the output to the input, facilitating the optimization process. The diagram below shows the architecture of ResNet that the researchers adopted for this research [6], [11].

Table 1. ResNet Architecture

Layers	Hyperparameters
Input layer	Grayscale Image
Conv1	F: 32, K: 3 x 3
Conv2	F: 64, K: 3 x 3
Pooling Layer	K: 2 x 2, S: 2 x 2
Block	F: 128, K: 3 x 3 F: 128, K: 3 x 3
Conv3	F: 128, K: 3 x 3
Pooling Layer	K: 2 x 2, S: 2 x 2
Block	F: 256, K: 3 x 3 F: 256, K: 3 x 3
Conv4	F: 256, K: 3 x 3
Pooling Layer	K: 2 x 2, S: 1 x 2
Block	F: 512, K: 3 x 3 F: 256, K: 3 x 3
Conv5	F: 512, K: 3 x 3
Block	F: 512, K: 3 x 3 F: 512, K: 3 x 3
Conv6	F: 512, K: 2 x 2 S: 1 x 2
Conv7	F: 512, K: 2 x 2 S: 1 x 2

Visual Geometry Group (VGG)

The Visual Geometry Group (VGG) architecture is known for its simplicity and effectiveness. It consists of a series of convolutional layers, followed by fully connected layers. The key characteristic of VGG is the use of small 3x3 convolutional filters stacked on top of each other, which allows for deeper networks while keeping the number of parameters manageable. The architecture typically starts with a few initial convolutional layers, each followed by a rectified linear unit activation function. The ReLU non-linearity helps introduce non-linearities into the network and improves its ability to learn complex patterns in the data. The following diagram is adapted from two variations of the architectures in [7], [11].

Table 2. VGGArchitecture

Layers	Hyperparameters
Input layer	Grayscale Image
Conv1	F: 64, K: 3 x 3
Pooling Layer 1	K: 2 x 2, S: 2 x 2
Conv2	F: 128, K: 3 x 3
Pooling Layer 2	K: 2 x 2, S: 2 x 2
Conv3	F: 256, K: 3 x 3
Conv4	F: 256, K: 3 x 3
Pooling Layer 3	K: 1 x 2, S: 2 x 2
Conv5	F: 512, K: 3 x 3
Batch Normalization	
Conv6	F: 512, K: 3 x 3
Batch Normalization	
Pooling Layer 4	K: 1 x 2, S: 1 x 2
Conv7	F: 512, K: 3 x 3 S: 1 x 2

Sequence Modelling

Sequence modeling in machine learning refers to the process of predicting or generating a sequence of outputs based on a sequence of inputs. In this case, the accepted input is the feature extraction output, where the output is the sequence to be fed into dense layer. The researchers also adopted this approach from [9]. In this study, the researchers utilized one of the most renowned RNN architectures, known as Bidirectional Long Short-Term Memory (BiLSTM), inspired by previous work done in [11].

Bidirectional Long Short-Term Memory is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. A LSTM uses three gates: an input gate, a forget gate, and an output gate for processing. Each gate has its own formula to compute whether to preserve or delete the information. In a standard LSTM, the hidden states at each time step are influenced only by the preceding context. However, in BiLSTM, there are two separate LSTM layers: one processing the input sequence in the forward direction and the other processing it in the backward direction. This allows the model to capture dependencies not only from the past but also from the future, enhancing its understanding of the temporal dynamics of the sequence.

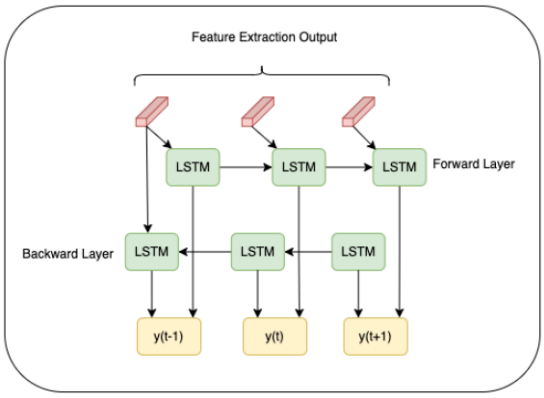


Fig. 2. Sequence Modelling with BiLSTM

Connectionist Temporal Classification

Cost function. In this research, the researchers utilized the cost function called Connectionist Temporal Classification (CTC) loss to calculate the difference in value between the ground truth and prediction texts.

The reason behind this is the research had to deal with sequence data where we need alignment between those sequences. The function itself calculates the cost by taking a continuous time series mapped to the target sequence. The loss function sums up the probabilities of all valid alignments that could produce the target sequence and the value is extracted by performing negative logarithms (also known as negative log loss) of the sum probabilities [8].

Decoder The main objective of CTC is given a set of characters called X and a set of all possible alignments of the characters called Y, is to find the alignment with the highest probability of being the right word. CTC does this by going through all the possible alignments, calculating the probability from one character to the next. [9] The algorithm itself also removed repeating blank token and characters as it went through all columns of the sequence modeling process.

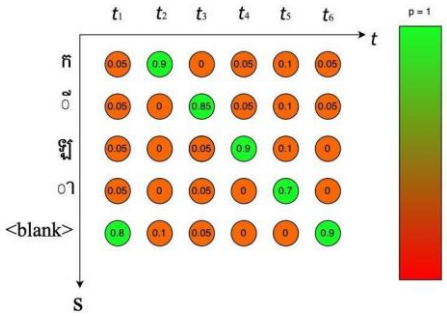


Fig. 3. CTC Greedy Decoding

Training Configurations and Process Followed by data preprocessing procedure, which involves morphological operation (dilation), binarization and label encoder to convert characters into integers. The next operation is features extraction, performed by the CNN models (ResNet, VGG) to produce a feature map. The feature map is resized into a shape of $Batch\ Size \times$ Sequence Length $\times\ Output\ Channel$, make it ready to pass into the sequence modeling stage, where BiLSTM captures the contextual information to be decoded into a sequence of character integers. To calculate the loss, a CTC loss function is used to calculate the error between the prediction and ground truth by performing the negative log loss on the probability, followed by weight adjustment from the optimizer (Adam with learning rate of 0.001). The training process begins with the images batching.

Table 3. Training Configurations

Hyperparameters	Value
Input layer	Grayscale Image
Hidden Size	256
BiLSTM number of layers	2
Optimizer	Adam
Batch Size	128
Loss Function	CTC Loss
Learning Rate	0.001

4 Evaluation Method

For this study, the researchers test the model on an artificially generated dataset using one primary metric: CER stands for Character Error Rate and is calculated by dividing the number of unrecognized characters by the total number of characters. [5] It shows the number of wrong predictions made by the model. The CER is defined by the following formula.

$$CER = \frac{(S+I+D)}{Number\ of\ Characters} \tag{1}$$

In this formula, S stands for Substitution, I stands for Insertion and D stands for Deletion. A Substitution is when a character gets replaced within a word for the word to be the correct word. An Insertion is when an extra character is added to an original word. A Deletion is when a character is removed from the actual word. CER is based on the Levenshtein distance which measures the difference between two strings character-wise [5].

5 Experimentation Result

5.1 ResNet + BiLSTM

Fig. 4a depicts the Train Loss, Validation Loss, while Character Accuracy Rate of the ResNet architecture is shown in Fig. 4b. The train loss represents the error between the predicted output of the model and the target output during the training phase of the model. The validation loss measures the error on the validation data set.

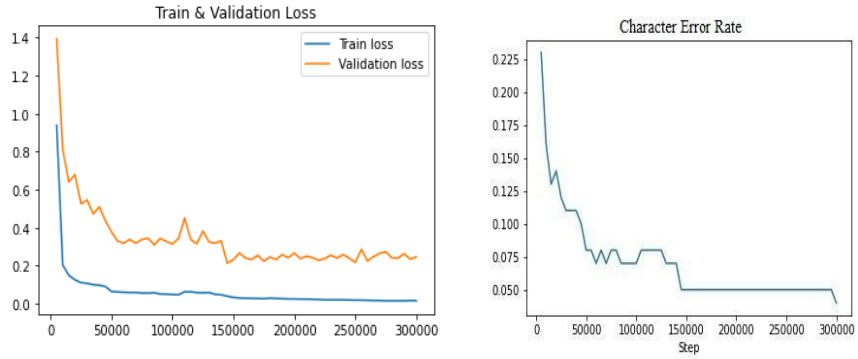
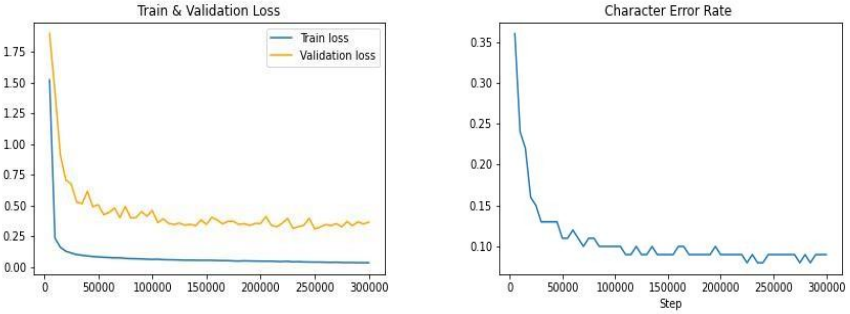


Fig. 4. (a) ResNet Train and Loss Validation Loss, (b) ResNet Character Error Rate

The Character Error Rate (CER) measures the discrepancy or error rate between the predicted or transcribed characters and the ground truth or reference characters as shown. As the number of steps or epochs increase in the x-axis, the train loss, validation loss, and character error rate decrease which is the desired outcome as it indicates how well the model is performing in the training and validation.

5.2 VGG + BiLSTM

Similarly, Fig. 5a depicts the Train and Validation Loss, while Character Error Rate of the VGG architecture is displayed in Fig. 5b. The same thing happens during training with VGG architecture, as it does with the ResNet architecture, where the training loss shows little improvement. What is noteworthy here is that over the same number of steps, the VGG architecture has a higher CER than the ResNet architecture.



(a) (b)

Fig. 5. (a) VGG Train and Validation Loss, (b) VGG Character Error Rate

5.3 Result

Despite the value of train loss starts to show little to no improvement after 200000 steps on both architectures, the decision to keep on training persists as the validation loss is still showing some improvement. This indicates that model is still learning and generalizing better on unseen data. The gap between the training and validation loss also suggests that the models are not overfitting. The plateau in training loss can be attributed to many factors and one of them is that the models have already captured the majority of pattern in the data and it leads to slower convergence.

As can be seen from the Table 4, both of the proposed models coming from the proposed method outperform Tesseract by roughly 20% accuracy rate, while ResNet outperforms VGG in this project as it provides a higher accuracy rate compared to VGG, despite having a minor trade-off in terms of speed. The character error rate between ResNet and VGG as feature extraction is shown in Fig. 6.

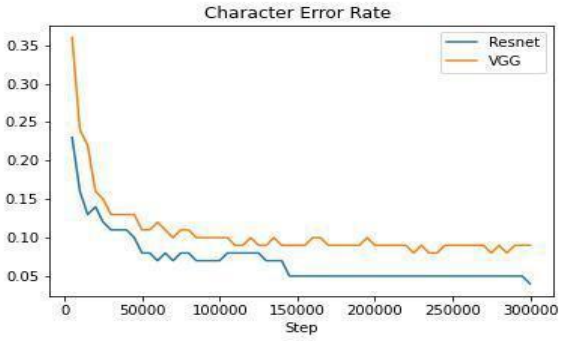


Fig. 6. Comparison Character Error Rate between ResNet and VGG

These results are based on our custom dataset, which includes the processing of image using white-pixel masking horizontally and vertically over a specific number of

iterations, further demonstrating the capability of the proposed models along with the image processing techniques in handling the partially broken text.

However, it is important to note that the proposed models can perform well under the partially broken form (salt and pepper manner). For example, highly broken characters where significant portion of characters are missing and disconnected, could present a greater challenge, and it could serve as an interesting topic for future work, requiring the model architecture adjustment to fulfill this particular challenge.

Table 4. Table captions should be placed above the tables.

Model	CAR	CER	Speed
ResNet + BiLSTM	96%	4%	75ms
VGG + BiLSTM	92%	8%	40ms
Tesseract	74%	26%	45ms

6 Conclusion

The study focuses on the development and optimization of an Optical Character Recognition (OCR) model, specifically aimed at improving the recognition of Khmer characters, including those that are partially broken.

Based on the findings, necessary data for model training and testing was generated using Python libraries. Dilation, an image processing technique, was employed to restore partially broken characters, enhancing the model's accuracy. Feature Extraction and Bidirectional Long Short-Term Memory (BiLSTM) networks were identified as critical factors influencing the OCR system's speed and accuracy. The model's effectiveness depends on the testing dataset quality and measurement techniques; without appropriate methods, the model's accuracy may not reflect real-world performance.

Recommendations for future research include generating Khmer broken characters using similar or alternative approaches, exploring other morphological operations or deep learning techniques like GANs for improved time efficiency and performance.

Acknowledgments. The researchers would like to extend their gratitude to Mr. Neil Ian C. Uy, Dr. Sa Math, and Dr. Hongly Va, for their constructive feedback and insightful guidance which have notably contributed to the improvement and success of this research.

Disclosure of Interests. It The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Conexiom: Advantages, Challenges, and Alternatives to OCR Solutions. https://conexiom.com/blog/the-advantages-challenges-and-alternatives-to-ocr-solutions (2024). Accessed 18 Jun 2024
2. Praseetha, M., Deepa, S.: Segmentation in Malayalam OCR - Handling Broken Characters Using Active Contour Model. In: International Conference on Control, Instrumentation, Communication and Computational Technologies, pp. 1-6. IEEE (2014)
3. Ramalingam, K., Bhojan, R.: Identification of Broken Characters in Degraded Documents. Int. J. Intell. Eng. Syst. 13(3), 130–137 (2017). https://oaji.net/articles/2017/3603-1524459510.pdf.
4. Sandhya, N., Krishnan, R.: Broken Kannada Character Recognition - A Neural Network-Based Approach. In: International Conference on Electrical, Electronics, and Optimization Techniques, pp. 1-5. IEEE (2016)
5. Buoy, R., Taing, N., Chenda, S., Kor, S.: Khmer printed character recognition using attention-based Seq2Seq network. Ho Chi Minh City Open University Journal Of Science-Engineering And Technology, 12(1), 3-16 (2022).
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv:1512.03385 (2015). https://arxiv.org/abs/1512.03385. Accessed 10 Dec 2015
7. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 (2015). https://arxiv.org/abs/1409.1556.
8. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In: Proceedings of the 23rd International Conference on Machine Learning (ICML), pp. 369–376. ACM, Pittsburgh, PA, USA (2006). https://www.cs.toronto.edu/~graves/icml_2006.pdf
9. Yao, C.: An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. arXiv:1507.05717 (2015). https://arxiv.org/abs/1507.05717
10. Yingjie, L., Fucheng, Y.: Application of Mathematical Morphology on Touching or Broken Characters Processing. Information & Mechanical Engineering School, Beijing Institute of Graphic Communication, Beijing, China (2010). https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6e74d839db14f61db6bbc053a4232b0f58238c6d.
11. Baek, J., et al.: What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. arXiv (2019). https://doi.org/10.48550/arxiv.1904.01906.