A Synthetic Recipe for OCR

David Etter*, Stephen Rawls†, Cameron Carpenter‡, Gregory Sell*

*Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, USA

†Information Science Institute
University of Southern California

detter2@jhu.edu srawls@isi.edu ccarpe18@jhu.edu gsell@jhu.edu

***Abstract*—Synthetic data generation for optical character recognition (OCR) promises unlimited training data at zero annotation cost. With enough fonts and seed text, we should be able to generate data to train a model that approaches or exceeds the performance with real annotated data. Unfortunately, this is not always the reality. Unconstrained image settings, such as internet memes, scanned web pages, or newspapers, present diverse scripts, fonts, layouts, and complex backgrounds, which cause models trained with synthetic data to break down. In this work, we investigate the synthetic image generation problem on a large multilingual set of unconstrained document images. Our work presents a comprehensive evaluation of the impact of synthetic data attributes on model performance. The results provide a recipe for synthetic data generation that will help guide future research.**

I. Introduction

Optical character recognition (OCR) performance on document images has increased significantly with the rise of neural network architectures [1] [2] [3]. Many of these advances have occurred in constrained settings [4], where the images are drawn from a single domain and language, such as book scans or scene text. The performance of these systems often deteriorates when presented with document images from an unconstrained setting that includes multiple domain and languages. Unconstrained document images can include maps, forms, web pages, and social media images (see Table I for examples). This challenging setting is often multilingual and can include text over complex backgrounds, multiple fonts, lighting changes, and occlusions, and systems unprepared for this diversity typically degrade in performance. As an example, Table V shows that the character error rate (CER) for the pre-trained Tesseract best LSTM models more than doubles when evaluated on our unconstrained Chinese test set.

One of the primary challenges with creating better models for the unconstrained scenario is the lack of annotated training data. Neural models require tens of thousands of annotated text line images that are both expensive and time consuming to produce.

Synthetic image generation has emerged as a solution for the OCR training data problem [5] [6]. Given a few fonts and some seed text, we can produce an unlimited number of training images. However, this generic approach to synthetic models breaks down when the attributes of the synthetic images do not match the test conditions.

In this work we investigate the synthetic image generation problem on a large multilingual set of unconstrained document images. We ask the questions that many researchers have faced when attempting to train an OCR model with synthetic data:

- How many fonts should we use?
- What styles or attributes need to be applied to the text?
- How many images are needed for training?
- What seed text should we use?

To answer these questions, we conduct detailed experiments in this unconstrained setting on two challenging languages: Chinese and Russian. Our results provide a recipe for synthetic data generation that will help guide other researchers.

Synthetic data has been used in training OCR systems [7] [8] [9] at the character, word, line, and document level. It has also been applied to a variety of OCR domains such as CAPTCHA [10], face recognition [11] [12], biomedical research [13], and scene text [14] [3] [15]. The synthetic tool developed by [14] was specifically designed to render scene text images. Their approach takes a scene text image and then uses the scene layout during text placement in order to render more realistic in-scene text.

The work of [16] provides one of the most detailed descriptions of the attributes used for synthetic data generation. They generate word-level images using a variety of fonts, colors, distortions, and image blending.

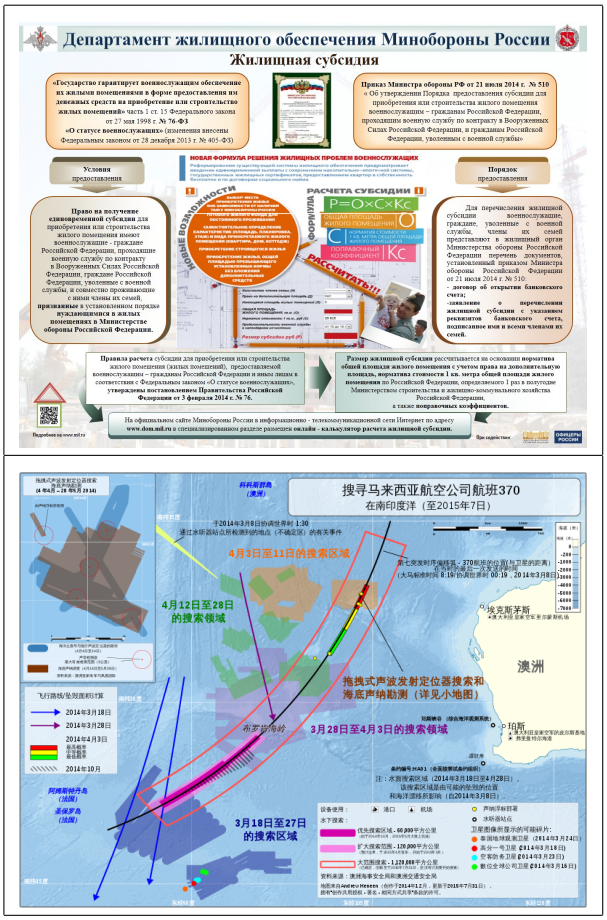
Synthetic generation has also been applied to document-level images in the work of [17]. Given a real document image, they apply a semi-automatic approach to extract font, background, and document layout. These attributes can then be used to generate realistic synthetic documents using random seed text.

Our work makes the following contributions to the OCR community:

1) First, we evaluate synthetic data generation on a large multilingual set of unconstrained document images.
2) Next, we develop a recipe for synthetic data generation

‡Work done during the HLTCOE SCALE workshop

TABLE I
EXAMPLE UNCONSTRAINED IMAGES (REUSE LICENSE)
RUSSIAN SOCIAL MEDIA (TOP), CHINESE DIAGRAM (BOTTOM)



based on a comprehensive evaluation of font, style, seed text, and quantity.

3) Finally, we provide a direct comparison of model performance for real vs synthetic training data.

II. APPROACH

A. Synthetic Generation

Most synthetic data generation tools render images at the word or line level. Given a list of seed text and a list of fonts, an image is generated for each seed input. This simple approach appears to work for many constrained text scenarios that contain uniform fonts and simple backgrounds. However, this approach begins to break down in unconstrained settings, such as with the complex layouts, diverse fonts, and backgrounds seen in the example images in Table I.

To overcome these limitations, we use a synthetic generation tool that has the ability to render multiple line images into complete documents, allowing us to construct templates that capture the complex document attributes not available in a simple line image. As an example, we can pad the bounding box around a given line and capture pieces of characters from lines above and below the current line. This provides a real document image artifact that occurs when the recognizer is run downstream of a text detector.

To create the data itself, the rendering engine of a web browser is used to convert HTML documents into annotated images. By constructing synthetic documents using CSS and HTML we can easily control layout, font, style, and backgrounds. The approach allows us to generate large numbers of synthetic documents with the types of variability that are found in real document images. JavaScript is used to extract the locations of text regions in the HTML document and create ground truth bounding box annotations.

B. System Description

To train our OCR system, we follow the setup of [2] [18] using an end-to-end neural model. This model combines convolutional neural networks (CNNs) for feature extraction with long-short term memory (LSTM) recurrent networks for sequence modeling.

We force a fixed image input height into the CNN-based feature extractor, which is based on the VGG architecture [19] that was used in ImageNet ILSVRC-2014 [20]. The network also includes fractional max-pooling layers [21] that are biased towards input width in order to keep more features in this dimension. The output from the final convolution layer is passed to a fully-connected layer, which reduces dimensionality before input into a stacked bidirectional LSTM.

Output from the LSTM layers is passed into a fully-connected layer, which matches the size of the input character set. We apply the connectionist temporal classification (CTC) loss [22] to allow for segmentation-free training and then use a greedy arg-max for decoding at test time.

C. Font Attributes

One of the key challenges for synthetic generation is identifying a set of fonts that match the diversity of the data. We investigate whether a small number of fonts can provide this coverage or if a breadth of fonts is required to match the characteristics of the data set. Our approach was to first train models using synthetic data generated using a single font and then incrementally increase the number of fonts for subsequent experiments.

We gathered hundreds of open-source fonts and then separated them by the supported language. One of the challenges with open-source fonts is that many support only a subset of the true character set and often do not follow standards for identifying glyph support. Our approach was to gather as many open-source fonts as possible and then render each unique character from our seed set to validate the content of the font packages. Table II shows the unique characters from the training set rendered with two different fonts. During synthetic generation, a random font is selected for each seed sentence, with a back-off font available for instances where a character glyph is not available.

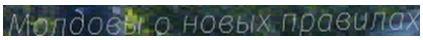
D. Style Attributes

Style defines the attributes that are used to render our seed text into synthetic images. This includes items such as font

TABLE II
FONTS: ALEGREYA-REGULAR (LEFT), LOBSTER-REGULAR (RIGHT)

Спасибо звнме!Ифрц-тдчкйл 8025(Ры)4963metodis@rfa.u ГАМОТНЬЧШЕЙКУЛЫ№1у,хгБщшя ь“”ВжДюХЮЗэЯЭПАр:«ѓ;ёЦЖl chn%KkZbg"REPLYVwB=?IФ$ъ/ ÉýЩzvy_·#T−CDM\|S*Жiьʒ7ЁIq NFJQ'UéXjHöWGx\−→·•áж• +ÿO…≠<>ó$Ó[]ø©ЪЂїΘ~&®' 'e°Р„əƏ˚	*Спасибо звнме!Ифрц-тдчкйл* *8025(Ры)4963metodis@rfa.u* *ГАМОТНЬЧШЕЙКУЛЫ№1у,хгБщшя* *ь“”ВжДюХЮЗэЯЭПАр:«ѓ;ёЦЖl* *chn%KkZbg"REPLYVwB=?IФ$ъ/* *ÉýЩzvy_·#T-CDM\|S*Жiьʒ7ЁIq* *NFJQ'UéXjHöWGx\−·áж•* *+ÿO…≠<>ó$Ó[]ø©ЪЂїΘ~&®'* *'e°Р„əƏ˚*

TABLE III
SYNTHETIC ATTRIBUTES

Font / Background Color	
Background image	
Pad	
Rotate	

color, background color, and background images. We also include text attributes, such as padding and rotation, into our style definition. The goal is to quantify the impact of each of these synthetic attributes on model training. Table III provides examples of 4 types of style applied to Cyrillic text.

E. Seed Attributes

Given the diversity of our training set, we investigate whether the underlying domain of the seed text can impact synthetic data training. The synthetic approach loads text from a source corpus and randomly selects a sentence for each image.

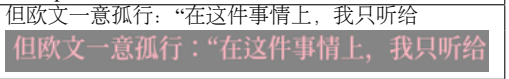
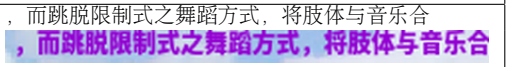
To generate synthetic images, we use seed text sentences drawn from news, web crawl, Wikipedia, and YOMDLE (defined below) annotations. Table IV provides examples of the different seed types. The news domain provides formal text sentences and often includes a more technical vocabulary. The web crawl is an informal corpus, which is often unstructured but covers a wide domain of topics. The Wikipedia corpus is a semi-formal corpus that includes both short facts and long descriptive sentences. This corpus also covers a wide domain and includes many technical terms.

III. DATA SETS

The evaluation data for this work is drawn from the SLAM data set, which was collected by the University of Maryland Center for Advanced Studies of Language (CASL). It includes at least 500 images for each of 33 languages, with 25 unique scripts. A subset of these images for five of the languages were boxed at the line image level and transcribed.

In our synthetic experiments, we focus on the Chinese and Russian languages, two of the transcribed languages, in order to demonstrate results on two distinct scripts and a large

TABLE IV
EXAMPLE CHINESE SEED TEXT AND SYNTHETIC IMAGE

Language	Example Seed Text
News	但欧文一意孤行：“在这件事情上，我只听给 
Web	，而跳脱限制式之舞蹈方式，将肢体与音乐合 
Wiki	时1813年7月27日出的日全食 

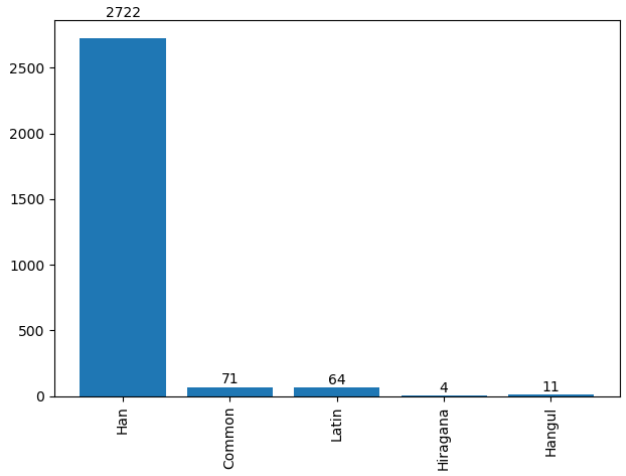


Fig. 1. SLAM Chinese character distribution

diverse set of characters. The Chinese evaluation set contains over 10 thousand annotated line images of simplified Chinese. Figure 1 provides a breakdown of the characters in the Chinese SLAM data set. The figure shows that we have over 2700 unique Han characters. These line images have an average height of 35 pixels and an average width of 230 pixels. The Russian evaluation set contains over 11 thousand annotated lines with an average height of 28 pixels and an average width of 310 pixels. Figure 2 provides a breakdown of the Russian character distribution. Examples of the types of images in this data set include scans of web pages, newspapers, receipts, phone books, forms, maps, menus, and social media captures. The text in these images provide a diverse set of fonts and complex backgrounds.

The training data used in our real data experiments is drawn from data gathered and transcribed by Yet One More Deep Learning Enterprise, which we will call the YOMDLE dataset. This data set is drawn from a similar domain as the SLAM set and includes scans of newspapers, web pages, presentations, and social media captures. The set has a slightly higher resolution than the SLAM set and includes over one thousand images with annotated and transcribed lines for each of 8 languages, including our evaluation languages of Russian and Chinese. The Chinese collection contains over 13 thousand transcribed line images with an average height of 52 pixels and

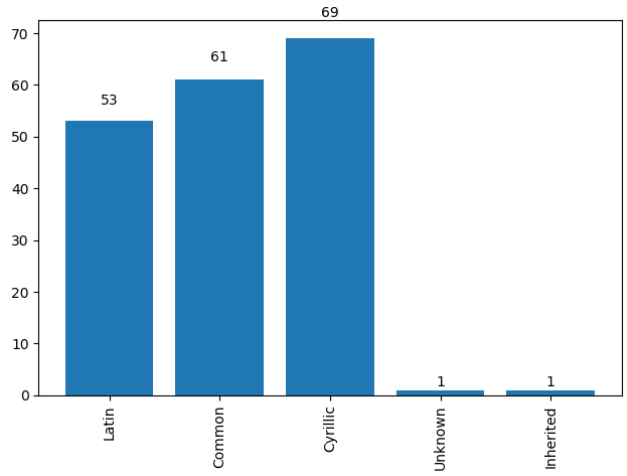


Fig. 2. SLAM Russian character distribution

an average width of 503 pixels. The Russian set contains over 14 thousand images with an average height of 45 pixels and an average width of 659 pixels.

The seed text for synthetic image generation is derived from either the annotations of real text images or corpora used by linguistic researchers. The real image seed text was taken from the transcribed YOMDLE text and images were generated using various fonts and backgrounds. To evaluate the contributions of various seed text domains, we also generated synthetic data using text gathered from news, web, and Wikipedia crawls. Each of the these linguistic data sets includes over 1 million sentences collected within the last 10 years.

IV. Experiments

We use the open source VistaOCR library[1] to conduct our experiments. The CNN-based system consists of 7 convolution layers. Each layer consists of a 2D-convolution with 3x3 filter kernels followed by Batch-Norm and RELU. A fractional max-pool layer occurs after layers 2 and 4, with the output ratio set at .5 for height and .7 for width. A fully connected layer provides a bridge to the 3-layer bidirectional LSTM and a final fully connected layer has an output size of the training alphabet size. The model is trained using warp CTC loss and an Adam optimizer. The initial learning rate is set to 1e-3 and is reduced by a factor of 10 when the loss plateaus. All images are resized to a fixed height of 30 pixels with a variable width that maintains aspect ratio.

At training time we randomly apply up to three augmentations to each image. These augmentation are generated using the ImgAug library[2] and include blur, noise, sharpen, emboss, pixel dropout, channel inversion, brightness, hue, saturation, contrast, and gray-scale. Models are trained with a mini-batch size of 32 images where batches are sorted in increasing image width order. All models are trained on NVIDIA 1080 GPU cards, using a single GPU. Results are evaluated using Word Error Rate (WER) for Russian and Character Error Rate (CER) for Chinese.

[1]https://github.com/isi-vista/VistaOCR
[2]https://github.com/aleju/imgaug

A. Baseline

As a comparison for the synthetic training experiments, baseline scores are established by training over real data. Language specific models are trained using the approximately 1K documents and 15K line images for the annotated YOMDLE Russian and Chinese data. During training, 10% of the line images are reserved for validation. Table V shows the baseline results for evaluation on the SLAM data sets. We also provide comparison results to the pre-trained Tesseract best LSTM trained models to show the challenges of this unconstrained setting. These numbers serve as a reference point for training with a reasonable amount of similar (YOMDLE) or mismatched (Tesseract) real images.

TABLE V
Baseline Results on SLAM data

Training data	Chinese (CER)	Russian (WER)
YOMDLE (Real)	13.3	9.1
Tesseract[3] (Best LSTM)	28.0	13.2

B. Style Attributes

In this set of experiments, font and style attributes are isolated to determine the impact on synthetic training data. The attributes used during the experiments are font count, font size, font color, background color, background images, padding, and rotation. For each experiment, all but one of the attributes is held fixed. The font attribute is selected from a list that includes over 60 fonts for Chinese and more than 600 Russian fonts. Font size is derived from a range of 12 to 24 point. Both font color and background color are selected from a standard color palette. The background image list includes over 30 thousand non-text images. For padding, a random pixel value between -1 and 6 is added to the image height and width. Finally, the text is rotated between -2 and +2 degrees.

For the font count experiments, models are trained using synthetic images rendered using 1, 5, and 60+ fonts. The experimental results are shown in Table VI. The single font used for both Russian and Chinese was a Noto Serif style font with regular thickness. The five font list includes fonts from both the Serif and Sans Serif families and both bold and regular thickness. The 60+ font list includes over 60 fonts for Chinese and more than 600 Russian fonts. As would be expected, the results show that more fonts are always better, but a somewhat surprising result is that the model still performs well when trained with as few as 5 fonts.

In the fixed font size experiment, a 12 point font is used for all training data and the remaining attributes are randomly selected, as described previously. The results in Table VII show that font size has a large impact on the overall model training.

TABLE VI
FONT RESULTS

Attribute	Chinese (CER)	Russian (WER)
1 Font	28.2	33.3
5 fonts	17.8	17.4
All fonts	16.3	12.7

The non-color experiment maintains a black font on a white background for all training images. This change causes a degradation of the model, since the test corpus includes a large number of images with complex backgrounds. However, somewhat surprising is that the removal of background images causes little change in the model performance.

TABLE VII
STYLE RESULTS

Attribute	Chinese (CER)	Russian (WER)
Fixed Font Size	23.0	19.7
No Color	21.1	16.9
No Background	17.2	13.0
All Styles	16.3	12.7

The removal of padding causes a large degradation in model performance, as seen in Table VIII. Bounding boxes in the evaluation set were annotated by multiple language experts and resulted in variations in the text cropping. This mirrors what would be expected from an upstream text detection analytic, and the results show simulating this effect in training is important for test-time performance.

Rotation did not affect the results as much as expected. Generally, rotating the training images is an augmentation that provides a good boost to the training process. In our scenario, the impact is decreased since the text detection process extracts bounding box sub regions and then axis aligns.

TABLE VIII
AUGMENTATION RESULTS

Attribute	Chinese (CER)	Russian (WER)
No Padding	23.0	14.0
No Rotation	19.2	12.5
All Augmentations	16.3	12.7

C. Seed Type Attribute

Seed type experiments help to determine if the domain of the seed text matters for synthetic model training. The web, news, and wiki seed text consists of 15K randomly selected lines from the corresponding language and domain corpora of the Leipzig Collection [23]. Models were trained using 30k images, created by using 2 instances from each seed line. The synthetic images were generated using all fonts, styles, padding, rotations, and online augmentation techniques. Our results in Table IX show that domain does have an impact on synthetic models. Presumably, the web domain outperforms models trained from news and wiki seed text. The informal nature of the web collection more closely matches the diverse test set collection of document images.

TABLE IX
SEED TYPE RESULTS

Training data	Chinese (CER)	Russian (WER)
Web	13.9	12.9
News	23.0	14.4
Wiki	16.4	15.1
YOMDLE	16.3	12.7

D. Instance Count

The instance count experiments look at how the number of generated instances impact a synthetic model. Given a seed set of approximately 15K lines images, can we improve the models by generating multiple instances of the same seed text.

In this experiment, the 15K YOMDLE language specific annotations are used as the seed text. Synthetic lines images are generated using 2, 10, and 30 instances of each line of seed text. Each instance is generated using a random selection of the 30+ fonts, font colors, background colors/images, crops, and rotations. The results in Table X show that increasing the number of instances from 2 to 10 has a small impact on the Russian data, but had a much larger impact on the Chinese data. There appears to be little gain from increasing the count beyond 10 instances. The largest decrease in error rate is for Chinese, which is a result of having additional training examples for the larger character set.

TABLE X
INSTANCE COUNT RESULTS

Training Count	Chinese (CER)	Russian (WER)
30K (2x)	16.3	12.7
150K (10x)	10.4	10.3
450K (30x)	9.1	10.0

V. CONCLUSIONS

We presented a recipe for synthetic image generation, shown in Table XI, that removes the guess work in how to train a purely synthetic model. Our approach is validated on a large diverse set of document images and on the challenging languages of Russian and Chinese. The results show that with a relatively small number of fonts and seed text, a model can be trained that is competitive or outperforms a model trained on real images.

TABLE XI
SYNTHETIC OCR RECIPE

Ingredients	Measurements
How many fonts should we use?	As little as 5 fonts will provide good performance, but more fonts are better.
What styles or attributes need to be applied to the text?	Padding, font color, and background color provide the largest impact.
How many images are needed for training?	30k - 150k lines will match or pass real data performance.
What seed text should we use?	Web data provides the most diverse domain.

In the future, we plan to employ this effective synthetic data training generation recipe to enable research in downstream natural language processing tasks such as named entity recognition and machine translation. In these cases, the existing annotated text can be used to seed the synthetic generation, resulting in images automatically labeled for the downstream task. Furthermore, we will explore extending the synthetic tool to model effects like motion blur in order to facilitate OCR in videos.

REFERENCES

[1] J. Wang and X. Hu, "Gated recurrent convolution neural network for ocr," in *Advances in Neural Information Processing Systems*, 2017, pp. 335–344.

[2] S. Rawls, H. Cao, S. Kumar, and P. Natarjan, "Combining convolutional neural networks and lstms for segmentation free ocr," in *Proc. ICDAR*, 2017. [Online]. Available: https://doi.org/10.1109/ICDAR.2017.34

[3] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 3304–3308.

[4] V. Margner and M. Pechwitz, "Synthetic data for arabic ocr system development," in *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001, pp. 1159–1163.

[5] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.

[6] T. A. Le, A. G. Baydin, R. Zinkov, and F. Wood, "Using synthetic data to train neural networks is model-based reasoning," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 3514–3521.

[7] X. Ren, K. Chen, and J. Sun, "A cnn based scene chinese text recognition algorithm with synthetic data engine," *arXiv preprint arXiv:1604.01891*, 2016.

[8] P. Krishnan and C. Jawahar, "Generating synthetic data for text recognition," *arXiv preprint arXiv:1608.04224*, 2016.

[9] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. Ramakrishnan, "Deep automatic license plate recognition system," in *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2016, p. 6.

[10] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *arXiv preprint arXiv:1312.6082*, 2013.

[11] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 293–303, 2018.

[12] A. Kortylewski, A. Schneider, T. Gerig, C. Blumer, B. Egger, C. Reyneke, A. Morel-Forster, and T. Vetter, "Priming deep neural networks with synthetic faces for enhanced performance," *arXiv preprint arXiv:1811.08565*, 2018.

[13] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 283–292, 2018.

[14] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[15] C. G. Serna and Y. Ruichek, "Classification of traffic signs: The european dataset," *IEEE Access*, vol. 6, pp. 78 136–78 148, 2018.

[16] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *arXiv preprint arXiv:1406.2227*, 2014.

[17] N. Journet, M. Visani, B. Mansencal, K. Van-Cuong, and A. Billy, "Doccreator: A new software for creating synthetic ground-truthed document images," *Journal of Imaging*, vol. 3, no. 4, 2017. [Online]. Available: http://www.mdpi.com/2313-433X/3/4/62

[18] S. Rawls, H. Cao, E. Sabir, and P. Natarajan, "Combining deep learning and language modeling for segmentation-free ocr from raw pixels," in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, April 2017, pp. 119–123.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[22] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[23] D. Goldhahn, T. Eckart, and U. Quasthoff, "Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages." in *LREC*, vol. 29, 2012, pp. 31–43.