Deep Learning-Based Detection of Penetration from Weld Pool Reflection Images

BY C. LI, Q. WANG, W. JIAO, M. JOHNSON, AND Y. M. ZHANG

ABSTRACT

An innovative method was proposed to determine weld joint penetration using machine learning techniques. In our approach, the dot-structured laser images reflected from an oscillating weld pool surface were captured. Experienced welders typically evaluate the weld penetration status based on this reflected laser pattern. To overcome the challenges in identifying features and accurately processing the images using conventional machine vision algorithms, we proposed the use the raw images without any processing as the input to a convolutional neural network (CNN). The labels needed to train the CNN were the measured weld penetration states, obtained from the images on the backside of the workpiece as a set of discrete weld penetration categories. The raw data, images, and penetration state were generated from extensive experiments using an automated robotic gas tungsten arc welding process. Data augmentation was performed to enhance the robustness of the trained network, which led to 270,000 training examples, 45,000 validation examples, and 45,000 test examples. A six-layer convolutional neural network trained with a modified mini-batch gradient descent method led to a final testing accuracy of 90.7%. A voting mechanism based on three continuous images increased the classification accuracy to 97.6%.

KEYWORDS

- Weld Pool
- Pool Oscillation
- Machine Learning
- Deep Learning
- Penetration
- Machine Vision
- Gas Tungsten Arc Welding (GTAW)
- Image

Introduction

Gas tungsten arc welding (GTAW) is one of the most widely used welding processes in industrial manufacturing, especially for critical applications, such as pressure vessels and aerospace. Benefits of GTAW include its stability and high-quality weld joints. In these critical applications, weld joint penetration is an important criterion to judge weld joint integrity, which affects mechanical properties especially fatigue and service life of the welded structure. Therefore, it is important to estimate the weld joint penetration in real time as an intermediate step to controlling the penetration in a desired state. The penetration state is determined by the depth and bottom surface of the weld pool, which are hidden from view and cannot be monitored directly in practical applications. Therefore, researchers have been trying to detect weld joint penetration using available characteristic information from the welding process.

The abrupt transition of the weld pool's natural oscillation frequency from incomplete joint penetration to complete joint penetration has been applied to monitor and control the weld joint penetration by Xiao and Ouden (Refs. 1, 2). Chen et al. found the depth of joint penetration was related to the infrared thermal images of the weld pool captured by infrared cameras, which were used for characterizing the surface temperature distribution of the weld pool (Refs. 3–5). Reflected ultrasonic waves have also been used to determine and control weld joint penetration in real time (Refs. 6, 7).

Recently, convolutional neural networks (CNNs) have become the prevalent method for solving computer vision problems. As a specialized kind of neural networks to process grid-like topology data, CNNs are inspired by the study of a monkey’s visual cortex (Refs. 20, 21). LeCun et al. proposed the architecture of modern CNNs (Ref. 22) and developed a five-layer CNN (LeCun-5), including two convolutional layers, two pooling layers, and one fully connected layer, which will be trained by the back propagation optimization algorithm (Ref. 23).

In this modern CNN architecture, convolutional layers are used to process raw pixels and extract features automatically; and fully connected layers are used to do high-level reasoning based on the features extracted. Inspired by the successful application of LeNet-5 in classifying handwritten digits, deeper and more ingenious CNNs have been developed to deal with more complex computer vision tasks. In 2012, AlexNet won the Large Scale Visual Recognition Challenge with a top five test error rate of 15.3%, compared to 26.2% achieved by the second-best entry using non-CNNs (Ref. 24). After the success of AlexNet, additional work has been done to improve the performance of CNNs for image classification, object detection,
and tracking, such as R-CNN (Refs. 25–27), ZFNet (Ref. 28), VGGNet (Ref. 29), GAN (Ref. 30), GoogLeNet (Ref. 31), and ResNet (Ref. 32). In addition, CNNs have also been successfully applied in other areas such as speech processing (Refs. 33, 34) and natural language processing (Refs. 35, 36). The analysis suggests that the ability of CNNs to directly process raw images avoids the problems of manual selection and extraction of features from the images.

This paper presents an effective CNN-based method to detect weld joint penetration from the raw reflected dot-structured laser pattern of a weld pool. A weld pool sensing system was designed and built to capture reflected dot-structured laser patterns and corresponding backside images of the joints simultaneously. Data augmentation was then performed to expand the size of the raw dataset, generating the following three independent datasets: training data, validation data, and test data. A six-layer convolutional neural network, including two convolutional layers, two pooling layers, one fully connected layer, and one regression layer, was trained taking the raw labels and using the revised mini-batch gradient descent method. The final test showed the prediction accuracy of the network layers is 90.7%.

This paper is organized as follows: Section I introduces the principles of CNNs; Section II describes the dataset building process; Section III presents the network architecture, hyper parameters, training process of the proposed model, testing results, and discussion; and Section IV summarizes this paper and draws conclusions.

I. Principle of Proposed Method

The basic components and framework of CNNs are similar across domain applications. Typically, the convolution operation is used in place of matrix multiplication in at least one of the network layers (Ref. 37). The network structure typically includes convolutional layers, pooling layers, and fully connected layers (Ref. 38). In convolutional layers, small-sized kernels flow across previous layers and operate via a dot product with the previous layer by

\[
S'(i, j, k) = \sum_{m,n} f^{-1}(i + m, j + n) K_k(m, n)
\]

where \( S' \) is the calculated value after convolutional layers; \( i, j \) indicates the position; \( f \) is the \( l \)th layer; and \( K_k \) is the \( k \)th kernel used in \( l \)th layer.

A non-linear activation function will take in the convoluted value to produce the next layer to extract the non-linear feature:

\[
f(i, j, k) = a(S'(i, j, k))
\]

where \( a \) is the activation function.

Typical activation functions are sigmoid, tanh, and ReLU (Refs. 39, 40). To decrease complexity and increase robustness of CNNs, pooling layers follow the convolutional layers. In pooling layers, a region of data from the last layer is compressed into one value using predefined pooling methods such as average pooling (Ref. 22), max pooling (Ref. 41), \( L_p \) pooling (Ref. 42), and stochastic pooling (Ref. 43):

\[
f(i, j, k) = \text{pool}(f^{-1}(m, n, k)), \forall (m, n) \in R_i
\]

where \( R_i \) is the local neighborhood around location \( (i, j) \).

After convolutional and pooling layers, traditionally fully connected layers are applied. In the fully connected layers, grid-like topology data in the previous layer is reshaped into a column vector, which is pre-multiplied by a weight matrix \( W \) and added with a bias term \( B_i \):

\[
f_i = W_i f_i^{-1} + B_i
\]

The classified labels are calculated using forward-propagation-based architecture and the parameters include kernels, weights, and biases that are denoted by \( \theta \) in CNNs. The average loss \( L \) is defined based on the difference between classified and true labels:

\[
L = \frac{1}{N} \sum_{n=1}^{N} l(0; y^{(n)}, o^{(n)})
\]

where \( y^{(n)} \) is the true label and \( o^{(n)} \) is the predicted label.

Typical loss functions include squared error, Hinge loss (Ref. 44), and cross-entropy loss (Ref. 37). The CNNs are...
trained using the stochastic gradient descent optimization technique. In our approach, the parameter updating rule was mini-batch stochastic gradient descent (mini-batch SGD), as shown in Equation 6:

$$\theta_{t+1} = \theta_t - \eta_t \nabla \theta \frac{L(\theta_t; y^{(i:n)}, o^{(i:n)})}{(6)}$$

where $\eta_t$ is the learning rate and $n$ is the mini-batch size. The method used to determine how the parameters are changed, from $\theta_t$ to $\theta_{t+1}$ toward the ones that can minimize Equation 5, in neural networks is called optimizer. When using Equation 6 to change the parameters, the optimizer is to adaptively adjust the learning rate $\eta_t$ based on the computed gradient to accelerate the training processes or achieve better prediction performance. Widely used optimizers include momentum (Ref. 45), Adagrad (Ref. 46), Adadelta (Ref. 47), and Adam (Ref. 48) that have been developed and became the benchmark works in training neural networks.

II. System Setup and Dataset Building

Weld Pool and Weld Joint Penetration Sensing System

In our experiments, autogenous, pulsed GTAW spot welds were conducted on a 0.125-in.-thick 304 stainless steel. In each pulsing period, the peak current was applied at 60 A for 47 ms and followed by a 20-A base current for 3 ms. The weld pool oscillated and an image was captured, one image per pulsing cycle, at 2 ms of the base current to ensure that the arc had become darker. A weld pool sensing system had been designed and established based on the experiment platform proposed in the literature (Ref. 15). As shown in Fig. 1, a 650-nm-wavelength 19 × 19 dot-matrix structured laser pattern was projected onto the weld pool surface at 30 deg from horizontal. On the other side, the reflected laser light was collected by a screen placed on the path of the reflected laser pattern. A Point Grey GZL-CL-22C5M-C high-speed camera (camera 1 in Fig. 1) with a 650-nm center wavelength band-pass optical filter captured the images from the screen at a speed of 1000 fps. At each base current period, the high-speed camera captured three images of the weld pool surface to use as the raw data of the neural networks. In addition, a Point Grey FL3-FW-03S1C with no optical filter (camera 2 in Fig. 1) captured one image of the backside of the weld bead at the start of the base current period, saved in 8-bit black-and-white format. Figure 2A shows a typical reflected laser pattern image captured by camera 1 and used as input into our model. During the welding process, the weld pool surface was not even, which caused the reflected laser pattern to be irregular. As discussed, all the information regarding the backside weld penetration were contained in that image and used as the input data to the convolutional neural network. Figure 2B is a typical image captured by camera 2 and used to identify weld penetration state labels after
determine the best threshold setting. Preliminary experiments were done to find lighting condition, for example, that may affect the results. Throughout the experiments, all the lighting conditions and camera settings remained the same. The reason for the threshold being set to 110 was to eliminate the factors lighting condition, for example, that may affect the backside light area. The original dataset of reflected laser patterns and weld penetration states included 3540 examples. This is insufficient to train CNNs, leading to overfitting as the number of parameters is often tens of millions or more. In Pinto et al. (Ref. 49), a simple V1-like model had been built, trained, and tested on this small dataset. However, when variations such as position changes or different sizes were added to the test set, the performance degraded dramatically. In this paper, a label-preserving data augmentation process was performed to increase the size of the dataset. The approach was similar to that in Ref. 24 where cropping original images into several patches was found to be an effective method for data augmentation (Ref. 50) and affine transformations, such as rotation and scaling for enlarging datasets.

Considering the fact that skilled welders are able to determine weld penetration states at different positions, distances, and orientations, the three geometric transformations of shifting, scaling, and rotation were used for transforming the raw input images. The data augmentation results are summarized in Table 2.

To make sure the testing results of the neural networks are accurate and will generalize, the training data set, validation set, and test set must be completely separate. To accomplish this, we randomly partitioned the data into 75% training, 12.5% validation, and 12.5% test sets, balancing the number of examples from each label category. This resulted in a training set, which contained 270,000 images; a validation set, which contained 45,000 images; and a test set, which contained 45,000 images being created.

### III. Model Training

#### Architecture

Convolutional neural networks are used for many different computer vision tasks. For simple tasks, such as image classification, five to ten layers may be enough (Refs. 23, 24). However, if the task is complex, such as image segmentation or object detection, the neural network often contains more than ten layers. Examples include GoogLeNet (Ref. 1) with 22 layers and VGG-16 (Ref. 29) with 16 layers. Using deeper CNNs for simple tasks is not recommended because they require longer training time and increase the risk of overfitting. Details of the network architecture have to be designed to meet the requirements from each individual task. In our paper, a six-layer CNN architecture, includ-
Two convolutional layers, two max-pooling layers, one fully connected layer, and one softmax regression layer, had been selected based on preliminary experimentation using the validation set data. As shown in Fig. 5, 75 kernels with a size of $5 \times 5$ were used in the first convolutional layer. A $2 \times 2$ max-pooling layer followed the first convolutional layer. Adding a max-pooling layer enhanced the robustness of the position variance and significantly reduced the computational cost of the neural network. The second convolutional layer contained 50 kernels with a size of $5 \times 5$ followed by a $2 \times 2$ max pooling. All convolution operations were implemented without adding. Then, $50 \times 9 \times 6$ data was reshaped to a column vector with a length of 2700 and followed with a fully connected layer of 500 neurons. Lastly, a softmax regression layer was used to calculate the predicted label ranging from zero to five.

Currently, many CNNs use the rectified linear unit (ReLU) function as an activation function. Compared with traditional activation functions, such as tanh and sigmoid, ReLU often converges faster (Ref. 24) and achieves better performance (Ref. 39, 51). Although tanh and ReLU achieved similar validation set performance in our experiments, we chose to use ReLU as the activation function for faster convergence and lower risk of saturation. The number of parameters used in this CNN is summarized in Table 3.

### Table 3 — Parameters Used in Proposed CNN

| Layers               | Number of Parameters  |
|----------------------|-----------------------|
| Convolutional-1      | $(5 \times 5 + 1) \times 75 = 1850$ |
| Pooling-1            | $(1 + 1) \times 75 = 150$ |
| Convolutional-2      | $(75 \times 5 \times 5 + 1) \times 50 = 93,800$ |
| Pooling-2            | $(1 + 1) \times 50 = 100$ |
| Fully connected      | $2700 \times 500 + 500 = 1,350,500$ |
| So softmax regression| $500 \times 6 + 6 = 3006$ |
| Total                | 1,449,506              |

In our experiments, all weights were initialized using a Gaussian distribution with a zero mean and 0.01 variance, and biases were set to zero. We used a modified Xavier initialization (Refs. 29, 40) rather than full Xavier initialization (Ref. 52) to help address convergence difficulties. The default learning rate created some initial saddle points, so the initialization was reduced further by 0.1. The result was a validation set error rate of 22.48%, which was higher than the Gaussian initialization with a 22.08% validation error rate on the same validation dataset.

The optimizer was a revised mini-batch gradient descent with a mini-batch size of 600. As a tradeoff of batch gradient descent and stochastic gradient descent, mini-batch gradient descent took a small size of training samples each time to calculate the negative gradient direction and adjusted parameters to minimize the cost function. Although the mini-batch gradient descent balanced the variance of convergence and computation efficiency, the speed of convergence was relatively slow. Thus, a momentum term was added to speed up the convergence speed (Ref. 53). Learning rate annealing was used to balance the training time and the training accuracy. The initial learning rate $\eta_i$ was 0.01 in our experiments, decreas-
Long term, the goal of this work is to support the implementation of this system to control weld joint penetration in an industrial production line setting.

References

1. Xiao, Y. H., and Ouden, G. D. 1990. A study of GTA weld pool oscillation. *Welding Journal* 69(6): 289-s to 293-s.
2. Xiao, Y. H., and Ouden, G. D. 1993. Weld pool oscillation during GTA welding of mild steel. *Welding Journal* 72(8): 428-s to 434-s.
3. Chen, W., and Chin, B. 1990. Monitoring joint penetration using infrared sensing techniques. *Welding Journal* 69(4): 181-s to 185-s.
4. Nagarajan, S., Banerjee, P., Chen, W., and Chin, B. A. 1992. Control of the welding process using infrared sensors. *IEEE Transactions on Robotics and Automation* 8(1): 86–93.
5. Nagarajan, S., Chen, W., and Chin, B. 1989. Infrared sensing for adaptive arc welding. *Welding Journal* 68(11): 462-s to 466-s.
6. Carlson, N., and Johnson, J. 1988. Ultrasonic sensing of weld pool penetration. *Welding Journal* 67(11): 239-s to 246-s.
7. Hardt, D., and Katz, J. 1984. Ultrasonic measurement of weld penetration. *Welding Journal* 63(9): 273-s to 281-s.
8. Mnich, C., Al-Bayat, F., Debrunner, C., Steele, J., and Vincent, T. 2004. In situ weld pool measurement using stereovision. *Japan USA Symposium on Flexible Automation*, Denver, Colo.
9. Richardson, R., and Gutow, D. 1983. Coaxial arc weld pool viewing for process monitoring and control. *Welding Journal* 63(3): 43–50.
10. Zhao, C., Richardson, I., Kenjeres, S., Kleijn, C., and Saldi, Z. 2009. A stereo vision method for tracking particle flow on the weld pool surface. *Journal of Applied Physics* 105(12): 123104. DOI: 10.1063/1.3143789
11. Zhao, D., Yi, J., Chen, S., Wu, L., and Chen, Q. 2003. Extraction of three-dimensional parameters for weld pool surface in pulsed GTA welding with wire filler. *Journal of Manufacturing Science and Engineering* 125(3): 493–503. DOI: 10.1115/1.1556400
12. Agapakis, J. E., and Bolstad, J. O. 1991. Vision sensing and processing system for monitoring and control of welding and other high-luminosity processes. *Optics, Illumination, and Image Sensing for Machine Vision V*, pp. 32–38. International Society for Optics and Photonics: Boston, Mass.
13. Kovacevic, R., and Zhang, Y. 1996. Sensing free surface of arc weld pool using specular reflection: Principle and analysis. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 210(6): 553–564. DOI: 10.1243/PIME_PROC_1996_210_154_02
14. Zhang, Y., Song, H., and Saeed, G. 2006. Observation of a dynamic specular weld pool surface. *Measurement Science and Technology* 17(6): L9. DOI: 10.1088/0957-0233/17/6/L02
15. Song, H., and Zhang, Y. 2008. Measurement and analysis of three-dimensional specular gas tungsten arc weld pool surface. *Welding Journal* 87(4): 85-s to 95-s.
16. Ma, X., and Zhang, Y. 2011. Gas metal arc weld pool surface imaging: modeling and processing. *Welding Journal* 90(5): 85-s to 94-s.
17. Zhang, W., Wang, X., and Zhang, Y. 2013. Analytical real-time measurement of a three-dimensional weld pool surface. *Measurement Science and Technology* 24(11): 115011. DOI: 10.1088/0957-0233/24/11/115011
18. Zhang, W. J., Zhang, X., and Zhang, Y. M. 2015. Robust pattern recognition for measurement of three dimensional weld pool surface in GTA. *Journal of Intelligent Manufacturing* 26(4): 659–676. DOI: 10.1007/s10845-013-0825-z
19. Zhang, W., Liu, Y., Wang, X., and Zhang, Y. 2012. Characterization of three-dimensional weld pool surface in GTA. *Welding
Advances in Neural Information Processing Systems, 2014.

20. Hubel, D. H., and Wiesel, T. N. 1968. receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology 195(1): 215–243. DOI: 10.1113/jphysiol.1968.sp008455

21. Fukushima, K., and Miyake, S. 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. Competition and Cooperation in Neural Nets, pp. 267-285.

22. LeCun, Y., et al. 1990. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems, pp. 396–404.

23. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11): 2278–2324. DOI: 10.1109/5.726791

24. Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, pp. 1097–1105.

25. Girshick, R., Donahue, J., Darrell, T., and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587.

26. Girshick, R. 2015. Fast r-cnn. Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448.

27. Ren, S., He, K., Girshick, R., and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, pp. 91–99.

28. Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. European Conference.

29. Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

30. Goodfellow, I., et al. 2014. Generative adversarial nets. Advances in Neural Information Processing Systems, pp. 2672–2680.

31. Szegedy, C., et al. 2015. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

32. He, K., Zhang, X., Ren, S., and Sun, J. 2016. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

33. Abdel-Hamid, O., Mohamed, A.-R., Jiang, H., Deng, L., Penn, G., and Yu, D. 2014. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 22(10): 1533–1545. DOI: 10.1109/TASSP.2014.2339736

34. Sainath, T. N., et al. 2015. Deep convolutional neural networks for large-scale speech tasks. Neural Networks 64: 39–48. DOI: 10.1016/j.neunet.2014.08.005

35. Kim, Y. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

36. Kalchbrenner, N., Grefenstette, E., and Blunsom, P. 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

37. Goodfellow, I., Bengio, Y., and Courville, A. 2016. Deep Learning. MIT Press.

38. Gu, J., et al. 2015. Recent advances in convolutional neural networks. arXiv preprint arXiv:1512.07108.

39. Maas, A. L., Hannun, A. Y., and Ng, A. Y. 2013. Rectifier nonlinearities improve neural network acoustic models. Proceedings of the International Conference on Machine Learning 30(1).

40. He, K., Zhang, X., Ren, S., and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034.

41. Bourou, Y.-L., and Cun, Y. L. 2008. Sparse feature learning for deep belief networks. Advances in Neural Information Processing Systems, pp. 1185–1192.

42. Hyvarinen, A., and Koster, U. 2007. Complex cell pooling and the statistics of natural images. Network: Computation in Neural Systems 18(2): 81–100.

43. Zeiler, M., and Fergus, R. 2013. Stochastic pooling for regularization of deep convolutional neural networks. Proceedings of the International Conference on Learning Representation (ICLR).

44. Szegedy, C., et al. 2015. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2672–2680.

45. Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural Networks 12(1): 145–151. DOI: 10.1016/S0893-6080(98)00116-6

46. Duchi, J., Hazan, E., and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12(7): 2121–2159.

47. Zeiler, M. D. 2012. ADADELTA: An adaptive learning rate method. arXiv preprint arXiv:1212.5701.

48. Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

49. Pinto, N., Cox, D. D., and DiCarlo, J. J. 2008. Why is real-world visual object recognition hard? PLoS Computational Biology 4(1): 27. DOI: 10.1371/journal.pcbi.0040027

50. Simard, P. Y., Steinkraus, D., and Platt, J. C. 2003. Best practices for convolutional neural networks applied to visual document analysis. Proceedings of the Seventh International Conference on Document Analysis and Recognition, p. 958: IEEE Computer Society.

51. Glorot, X., Bordes, A., and Bengio, Y. 2011. Deep sparse rectifier neural networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 315–323.

52. Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256.

53. Karpathy, A. 2016. C231n: Convolutional neural networks for visual recognition. Neural Networks, Vol. 1.

54. Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning 4(2): 26–31.

55. Ioffe, S., and Szegedy, C. 2015. Batch normalization: Reducing internal covariate shift. International Conference on Machine Learning, pp. 448–456.

56. Wiesler, S., and Ney, H. 2011. A convergence analysis of log-linear training. Advances in Neural Information Processing Systems, pp. 657–665.

57. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1): 1929–1958.

58. Long, J., Shelhamer, E., and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.