Kaggle Report: An Application of a Convolutional Neural Network in Identifying Solar Photovoltaic Panels in Aerial Imagery

Team C – Coyotes: ***Frank Xu, Alan Zhou, Abhishek Angadi, Allison Young***

Abstract:

Data scientists are increasingly recognizing the value of deep learning methods for identifying objects in images. In this case, we developed an approach for identifying solar panels in aerial data using a pre-trained Convolutional Neural Network (CNN) model, MobileNet. The model trained on a labeled set of images labeled as either containing or not containing solar photovoltaic (PV) panels. Custom class-weights for the sigmoid loss function were utilized to offset an imbalance in the training data (which heavily preferred images without solar panels). The resulting model predicted correct labels from a publicly available test dataset and a blind test dataset, with an accuracy of 97.8% and 99.9%, respectively. While the light weight nature of the model may have limited test performance in comparison to other more sophisticated CNN models, the tradeoff allows the model to run on mobile devices, a useful trait for energy research.

1 Introduction

Solar energy provides a clean energy alternative to existing methods of electricity production. Byproducts of existing fossil fuel methods result in mass quantities of greenhouse gases and pollution that negatively impact the planet's climate, habitat, wildlife, and public health[1]. While the benefits of a transition to solar energy are well documented, one of the major challenges to implementing a transition from fossil fuels to renewable energy is understanding the market and distribution of small scale solar photovoltaic panels. These small panels, often privately owned, are a movement towards a shift in the overall power grid. However, it is a challenge to obtain complete and accurate information on where these solar panels are located, how many there are, and how much energy that may generate. Surveys and other data collection methods have been

[1] Jiangye Yuan et al., "Large-Scale Solar Panel Mapping from Aerial Images Using Deep Convolutional Networks," in *2016 IEEE International Conference on Big Data (Big Data)* (2016 IEEE International Conference on Big Data (Big Data), Washington DC,USA: IEEE, 2016), 2703–8, https://doi.org/10.1109/BigData.2016.7840915.

unsuccessful in producing a dataset sufficiently complete for use in policy and financial decisions regarding the integration of solar panels into the power grid at a broader scale[2].

This is where machine learning methods come into play. Utilizing aerial satellite data, computer vision algorithms are capable of learning and identifying important features and qualities from the appearance of solar panels in images. The algorithm, when deployed on new data, may determine the presence or absence of a solar panel within "unseen" imagery. These learning techniques, once refined, allow energy leaders to more accurately and efficiently identify appropriate solutions that guide the future design of energy production across the planet.

Machine learning provides a highly scalable solution to an otherwise tedious and manual process that is both costly and time consuming to implement. These data may also be used to identify information about the factors that correlate with the adoption of solar panel installation and utilization for individuals, neighborhoods, and broader communities. This research helps the industry understand its market and the methods necessary to progress towards clean and renewable energy production as a normative and common resource scalable to an entire nation[3].

2 Background

Extracting information from images is a difficult problem with many practical applications. Aerial imagery, however, contains massive amounts of information. Many researchers have used machine learning skills to detect or classify objects from aerial imagery. Mnih and Hinton demonstrated a method of detecting roads from aerial imagery using unsupervised learning models and achieved better results than traditional approaches on two urban datasets[4]. Graesser et al. used a Decision Trees Classifier to settle boundaries of formal and informal landscape settlements based on satellite imagery and achieved an overall accuracy from 85% to 92% in

[2] Jordan M. Malof et al., "Automatic Detection of Solar Photovoltaic Arrays in High Resolution Aerial Imagery," *Applied Energy* 183 (December 2016): 229–40, https://doi.org/10.1016/j.apenergy.2016.08.191.
[3] Stanford University, "Inventory Indicates Who Goes Solar and Why," Stanford News, December 19, 2018, https://news.stanford.edu/2018/12/19/inventory-indicates-goes-solar/.
[4] Volodymyr Mnih and Geoffrey E. Hinton, "Learning to Detect Roads in High-Resolution Aerial Images," in *Computer Vision – ECCV 2010*, ed. Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Lecture Notes in Computer Science (Springer Berlin Heidelberg, 2010), 210–23.

different areas[5]. Aguilar et al. used the Nearest Neighbor Classifier to classify the land-cover information in urban area and achieved over 91% accuracy[6]. Cheriyadat used unsupervised feature learning to prove that high-resolution satellite imagery can be used to automatically detect large facilities such as a shopping mall[7].

In more recent years, deep learning techniques have been used to identifying objects in aerial images. Cheng et al. used discriminative CNNs to boost the performance of remote sensing image scene classification, outperformed existing baseline methods and achieved state-of-the-art results on all test data sets[8]. In building a solar photovoltaic image dataset, Bradbury et al. also mentioned that modern classification techniques such as CNNs are widely used to train on labeled image data and detect objects[9]. In this case, we use a transfer-learning CNN model to train on our data and make predictions, as well as compare the performance with a base line model and evaluate performance.

3 Data Overview

As an image classification problem, we will have to train our model with a training dataset that consists of 1500 aerial images of roofs/solar panels and the labels indicating whether or not the

[5] J. Graesser et al., "Image Based Characterization of Formal and Informal Neighborhoods in an Urban Landscape," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, no. 4 (August 2012): 1164–76, https://doi.org/10.1109/JSTARS.2012.2190383.
[6] M. A. Aguilar, M. M. Saldaña, and F. J. Aguilar, "GeoEye-1 and WorldView-2 Pan-Sharpened Imagery for Object-Based Classification in Urban Environments," *International Journal of Remote Sensing* 34, no. 7 (April 10, 2013): 2583–2606, https://doi.org/10.1080/01431161.2012.747018.
[7] A. M. Cheriyadat, "Unsupervised Feature Learning for Aerial Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing* 52, no. 1 (January 2014): 439–51, https://doi.org/10.1109/TGRS.2013.2241444.
[8] G. Cheng et al., "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Transactions on Geoscience and Remote Sensing* 56, no. 5 (May 2018): 2811–21, https://doi.org/10.1109/TGRS.2017.2783902.
[9] Kyle Bradbury et al., "Distributed Solar Photovoltaic Array Location and Extent Dataset for Remote Sensing Object Identification," *Scientific Data* 3 (December 6, 2016): 160106, https://doi.org/10.1038/sdata.2016.106.

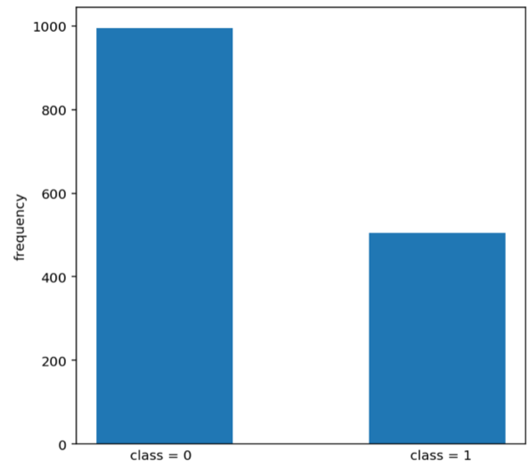
picture has a solar panel (Figure 1). The test data set was divided into two parts, one for the public leaderboard (about 508 images), one for the private leaderboard (about 762 images).

(Figure 1: Two examples from the training data set; left one is labeled as 0, while right one is labeled as 1)

The pictures all have a size of 101 × 101 pixels, and consist of three panels of R, G, and B. Therefore, there are 30603 features (predictors) for the label's binary classification (101 × 101 × 3). Each predictor from 0~ 255, with a relatively small (1500) size of training data. This is a typical phenomenon called Curse of dimensionality (number of predictors >> number of observations); when the dimensionality increases, the volume of the space increases exponentially and data become sparse. There are ways to break the curse; Loth et al. demonstrated its permeability using LASSO in regression to break the curse of dimensionality[10]. By the operation of convolution, we can also compress the large number of features into a smaller number of features.

[10] M. Loth, M. Davy, and P. Preux, "Sparse Temporal Difference Learning Using LASSO," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007, 352–59, https://doi.org/10.1109/ADPRL.2007.368210.

Regarding the data balance, out of 1500 training data, 505 of them are labeled as 1 (containing a solar panel), 995 of them are labeled as 0. Meaning, only 33.7% of the training data are labeled 1 (Figure 2). This can be concerning if the test data has a different distribution (e.g. more labeled 1 data than labeled 0). As a CNN, our model has no built-in functions to automatically deal with the imbalanced data; this situation will be further considered in the Methods section.

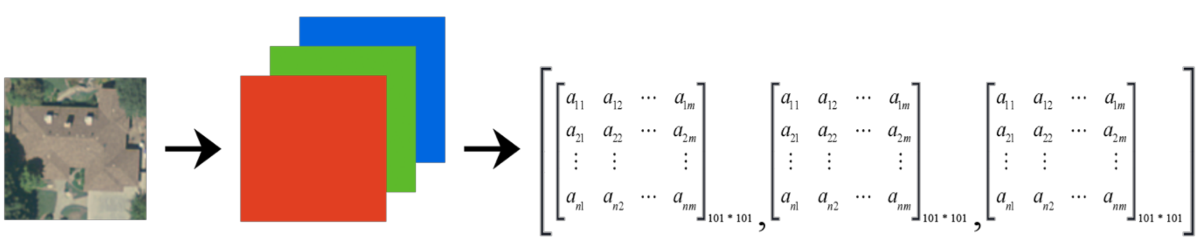


(Figure 2: Histogram of Imbalanced Training Data Distribution)

4 Methods

4.1 Data Reading

The training and testing images are stored in separate folders, labeled as *number.tif. We used a for loop to read in the images, as well as transform them into matrices (3 matrixes of 101 × 101 for each image) and concatenated the new matrix in the X_train matrix used for training (Figure 3). Therefore, we could have a 4-D array (1500 pictures × 3 matrixes × 101 pixels × 101 pixels), storing the training data. The testing data have a similar transformation.



(Figure 3: Turning a picture into feature matrices (3*101*101))

The training labels are stored in a .csv file, with a binary indicator (0 or 1) indicating the labels for each picture. We read the csv file as a 1-D array.

4.2 Data Pre-processing

Since our training data are automatically transformed from images, there is no need to check for missing values and scaling issues: all features (pixels) are a value ranged from 0 to 255. Given the fact that the solar panel can actually appear in every corner of each image, all the features are equally useful, and we cannot reduce dimensionality by abandoning features.

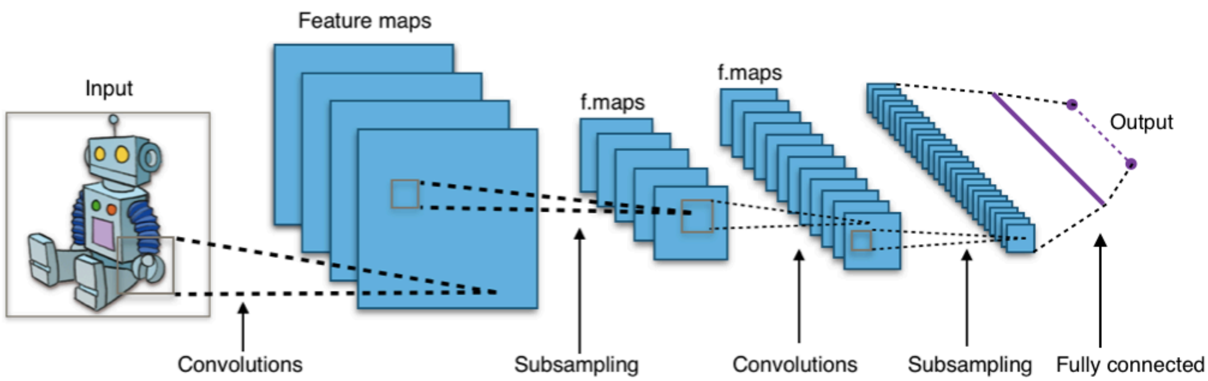
4.3 Model Selection

Image classification is a process in which we are given a set of images that are all labeled with a single category. We are then asked to predict these categories for a novel set of test images and measure the accuracy of the predictions[11]. Before CNN appeared, numerous methods were used by researchers for image classification, such as K-Nearest-Neighbors (KNNs), Support Vector Machines (SVMs), and Neural Networks (NNs)[12]. Compared with the previous methods, deep learning methods show notable potential in visual feature learning. One of the main foci of these methods is designing suitable deep network structures to accomplish specific tasks[13].

CNNs are one of the most popular categories of neural network deep learning, especially for high-dimensional data like images and videos for both supervised and unsupervised learning[14]. In 1990, LeCun et al. published the seminal paper establishing the modern framework of CNN, the back-propagation networks to handwritten digit recognition[15]. They developed a multi-layer

[11] "CS231n Convolutional Neural Networks for Visual Recognition," accessed March 4, 2019, http://cs231n.github.io/classification/.
[12] Xiao-Xiao Niu and Ching Y. Suen, "A Novel Hybrid CNN–SVM Classifier for Recognizing Handwritten Digits," *Pattern Recognition* 45, no. 4 (April 1, 2012): 1318–25, https://doi.org/10.1016/j.patcog.2011.09.021.
[13] Qin-Qin Tao et al., "Robust Face Detection Using Local CNN and SVM Based on Kernel Combination," *Neurocomputing*, SI: Recent Advances in SVM, 211 (October 26, 2016): 98–105, https://doi.org/10.1016/j.neucom.2015.10.139.
[14] Salman Khan et al., "A Guide to Convolutional Neural Networks for Computer Vision," *Synthesis Lectures on Computer Vision* 8, no. 1 (February 13, 2018): 1–207, https://doi.org/10.2200/S00822ED1V01Y201712COV015.
[15] Yann LeCun et al., "Handwritten Digit Recognition with a Back-Propagation Network," in *Advances in Neural Information Processing Systems 2*, ed. D. S. Touretzky (Morgan-Kaufmann, 1990), 396–404, http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf.

artificial neural network called LeNet-5 which could classify handwritten digits[16]. In 2012, Krizhevsky et al. trained a large, deep CNN model to classify 1.2 million high-resolution images into 1000 different classes with top 1 and top 5 error rates of 37.5% and 17.0%, which is considerably better than previous methods[17]. This is when CNN started to show its advantages in image classification, as it can be thought of as automatic feature extractors from images.



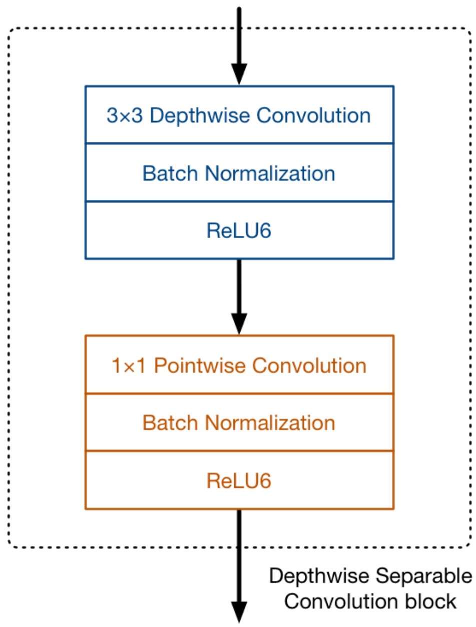
(Figure 4: A Typical CNN architecture, from Wikipedia[18])

Therefore, we decided to use Convolutional Neural Network to train and predict the data (Figure 4). We used a sequential model from the Keras package to build the model. We also adopted a pre-trained model structure and weights to further train on our own data, since our data size is relatively too small. This is called Transfer Learning, which as Pan et al. demonstrated, can greatly improve the performance of learning when we only have sufficient training data in another domain of interest, where the data may be in a different feature space or follow a different data distribution[19].

---

[16] Jiuxiang Gu et al., "Recent Advances in Convolutional Neural Networks," *Pattern Recognition* 77 (May 1, 2018): 354–77, https://doi.org/10.1016/j.patcog.2017.10.013.
[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, ed. F. Pereira et al. (Curran Associates, Inc., 2012), 1097–1105, http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.
[18] "Convolutional Neural Network," *Wikipedia*, February 27, 2019, https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=885403942.
[19] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering* 22, no. 10 (October 2010): 1345–59, https://doi.org/10.1109/TKDE.2009.191.

Presented by Howard et al. in 2017, MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks[20], which implies that it is faster than regular CNN models while retaining its accuracy. The full architecture of MobileNet consists of a regular 3×3 convolution as the very first layer, followed by 13 times the following building block (Figure 5). The activation function used here is ReLU6, which is similar to well-known ReLU, but restricts the maximum activation from becoming too big (larger than 6).



(Figure 5: A Depth-wise Separable Convolution building block, from machinethink.net[21])

We will use MobileNets as the pre-trained model, removing the last layer and adding a neural network layer which uses the sigmoid function and thus only deals with binary classification.

4.4 Data Fitting

As mentioned in the Data Overview section, there is imbalance with classes within the dataset, and CNN itself can be affected by this kind of imbalance. As Yue proposed in 2017, a simple yet effective way to address this imbalance could be to employ the weighted loss function as the

[20] Andrew G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *ArXiv:1704.04861 [Cs]*, April 16, 2017, http://arxiv.org/abs/1704.04861.
[21] "DepthwiseSeparableConvolution@2x.Png 544×724 Pixels," accessed March 4, 2019, https://machinethink.net/images/mobilenet-v2/DepthwiseSeparableConvolution@2x.png.

final layer of deep learning[22]. We also choose to adopt a class weight adjustment to offset the imbalance in the training data. A class weight helps the model to treat images of different classes differently, as we want the model to treat the class 1 data with more "weight". The adjustment of class weights is shown below:

$$W_{class=1} = \frac{Number\ of\ class = 0\ obs.}{Number\ of\ class = 1\ obs.}$$

$$W_{class=0} = 1$$

We use a proportion of 0.2 as the validation in each epoch, with a total 30 epoch, which means a total of $30 \times (1 - 0.2) \times 1500 = 36000$ pictures trained on the model. The batch size is 50.

5 Results

5.1 Performance Assessment

The CNN architectures mentioned above were constructed through the Keras library. Classification experiments were performed on a MacBook Pro with a 2.3 GHz Intel Core i5 CPU, 16GB RAM. The classification performance was assessed quantitatively by cross validation with an accuracy of 0.9533 and loss of 0.4611 under 30 epochs. The MobileNets model we used has built-in cross validation, auc score and f1 score. With 30 epochs, our auc score for validation data was 0.9845 and f1 score was 0.9427. This shows that our model is doing a great job predicting the validation data.

Finally, our model's prediction on 40% of test data reached an accuracy of 0.97789 in private leaderboard. Moreover, on 100% of test data, our test accuracy is 0.99871 which is close to 1 and outperformed most of our competing teams. The difference between the accuracy of the winning team and our team is only 0.00124.

[22] Songqing Yue, "Imbalanced Malware Images Classification: A CNN Based Approach," *ArXiv:1708.08042 [Cs, Stat]*, August 26, 2017, http://arxiv.org/abs/1708.08042.

5.2 Comparison to baseline model

We also compared our CNN model with a baseline model. The baseline model we chose was a simple logistic regression. The training accuracy was 0.719 and the validation accuracy was 0.702 (validation size = 0.2). The accuracy of the baseline model was 0.724 on the public testing data and 0.70667 on the private (blind) testing data.

The CNN model (MobileNet) outperformed the logistic regression on every data set. The model (accuracies ranging between 0.96 and 0.9987) has an increase of 37% in accuracy on average, in comparison to the base line model (Table 1).

Data Set	Base Model	Final Model – CNN (Mobile Net)
Training Accuracy	0.7190	0.9975
Validation Accuracy	0.7020	0.9600
Public Testing Accuracy	0.7720	0.9778
Private Testing Accuracy	0.7070	0.9987

(Table 1: The Comparison between Base Model and Final Model)

6 Conclusions

By training our transferred CNN model (MobileNet) with a class-weighted sigmoid loss function for 30 epochs, we had a training accuracy of around 0.9975 and validation accuracy of around 0.9600 (validation size = 0.2). Our predicted result for public test data (accounted for 40% of total testing data) achieved an accuracy of 0.97789, placing 8^{th} in the rankings. However, our predicted result for private testing data (accounted for 60%) achieved an accuracy of 0.99871, moving our ranking to 3^{rd}.

The similar accuracy from training/validation set and private/public testing set shows that there are no signs of overfitting. On the other hand, the huge leap of rank between public and private test set indicates that our model outperformed a majority of other teams when it comes to new testing data with slightly different distribution. We can further deduce that our treatment on class-weights had a positive impact on the model's overall performance.

However, our private testing result were still about 0.12% lower than the best score achieved. Given that we have trained our model for sufficient epochs, we believe that the major cause is that we trade some accuracy for speed. MobileNet is a light weight CNN model designed for mobile devices, which implies that it might sacrifice some accuracy for speed. Our model's training time for each epoch (1200 obs.) is only around 38s with our own laptop environment, which is considerably faster than most other pre-trained CNN models, as well as other machine learning methods.

7 Roles

All Coyote team members were active in the progressive steps of this competition from participation in team meetings, to understanding and implementing models, to compiling our work in a final report. Each member also played important individual roles throughout the project. Frank Xu identified potential models for exploration in this project from existing literature. The pre-trained model utilized in this competition was based on Frank's research. Abhishek Angadi ran a logistic regression model as an initial model and baseline for comparison to our final models, in addition to assisting in layer modification of the final models. Alan Zhou ran and troubleshooted a baseline KNN model as well as a second CNN model, designed from scratch. Ultimately, while Alan's CNN model was functional, the team decided to focus efforts on refining and developing the MobileNet model Frank identified because of its superior performance. Allison Young identified additional images for training the model and performed a literature review of existing methods, while also assisting in the paper composition and editing. All members participated in the refining of the model, with Frank taking lead on these tasks. In composing the report, all members identified relevant analyses for inclusion in the report, such as performance evaluation metrics and graphic explanations, and contributed to the written paper content.

References:

Aguilar, M. A., M. M. Saldaña, and F. J. Aguilar. "GeoEye-1 and WorldView-2 Pan-Sharpened Imagery for Object-Based Classification in Urban Environments." *International Journal of Remote Sensing* 34, no. 7 (April 10, 2013): 2583–2606. https://doi.org/10.1080/01431161.2012.747018.

Bradbury, Kyle, Raghav Saboo, Timothy L. Johnson, Jordan M. Malof, Arjun Devarajan, Wuming Zhang, Leslie M. Collins, and Richard G. Newell. "Distributed Solar Photovoltaic Array Location and Extent Dataset for Remote Sensing Object Identification." *Scientific Data* 3 (December 6, 2016): 160106. https://doi.org/10.1038/sdata.2016.106.

Cheng, G., C. Yang, X. Yao, L. Guo, and J. Han. "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs." *IEEE Transactions on Geoscience and Remote Sensing* 56, no. 5 (May 2018): 2811–21. https://doi.org/10.1109/TGRS.2017.2783902.

Cheriyadat, A. M. "Unsupervised Feature Learning for Aerial Scene Classification." *IEEE Transactions on Geoscience and Remote Sensing* 52, no. 1 (January 2014): 439–51. https://doi.org/10.1109/TGRS.2013.2241444.

"Convolutional Neural Network." *Wikipedia*, February 27, 2019. https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=885403942.

"CS231n Convolutional Neural Networks for Visual Recognition." Accessed March 4, 2019. http://cs231n.github.io/classification/.

"DepthwiseSeparableConvolution@2x.Png 544×724 Pixels." Accessed March 4, 2019. https://machinethink.net/images/mobilenet-v2/DepthwiseSeparableConvolution@2x.png.

Graesser, J., A. Cheriyadat, R. R. Vatsavai, V. Chandola, J. Long, and E. Bright. "Image Based Characterization of Formal and Informal Neighborhoods in an Urban Landscape." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, no. 4 (August 2012): 1164–76. https://doi.org/10.1109/JSTARS.2012.2190383.

Gu, Jiuxiang, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, et al. "Recent Advances in Convolutional Neural Networks." *Pattern Recognition* 77 (May 1, 2018): 354–77. https://doi.org/10.1016/j.patcog.2017.10.013.

Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." *ArXiv:1704.04861 [Cs]*, April 16, 2017. http://arxiv.org/abs/1704.04861.

Khan, Salman, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. "A Guide to Convolutional Neural Networks for Computer Vision." *Synthesis Lectures on Computer Vision* 8, no. 1 (February 13, 2018): 1–207. https://doi.org/10.2200/S00822ED1V01Y201712COV015.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, 1097–1105. Curran Associates, Inc., 2012. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LeCun, Yann, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. "Handwritten Digit Recognition with a Back-

Propagation Network." In *Advances in Neural Information Processing Systems 2*, edited by D. S. Touretzky, 396–404. Morgan-Kaufmann, 1990. http://papers.nips.cc/paper/293-handwritten-digit-recognition-with-a-back-propagation-network.pdf.

Loth, M., M. Davy, and P. Preux. "Sparse Temporal Difference Learning Using LASSO." In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 352–59, 2007. https://doi.org/10.1109/ADPRL.2007.368210.

Malof, Jordan M., Kyle Bradbury, Leslie M. Collins, and Richard G. Newell. "Automatic Detection of Solar Photovoltaic Arrays in High Resolution Aerial Imagery." *Applied Energy* 183 (December 2016): 229–40. https://doi.org/10.1016/j.apenergy.2016.08.191.

Mnih, Volodymyr, and Geoffrey E. Hinton. "Learning to Detect Roads in High-Resolution Aerial Images." In *Computer Vision – ECCV 2010*, edited by Kostas Daniilidis, Petros Maragos, and Nikos Paragios, 210–23. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010.

Niu, Xiao-Xiao, and Ching Y. Suen. "A Novel Hybrid CNN–SVM Classifier for Recognizing Handwritten Digits." *Pattern Recognition* 45, no. 4 (April 1, 2012): 1318–25. https://doi.org/10.1016/j.patcog.2011.09.021.

Pan, S. J., and Q. Yang. "A Survey on Transfer Learning." *IEEE Transactions on Knowledge and Data Engineering* 22, no. 10 (October 2010): 1345–59. https://doi.org/10.1109/TKDE.2009.191.

Tao, Qin-Qin, Shu Zhan, Xiao-Hong Li, and Toru Kurihara. "Robust Face Detection Using Local CNN and SVM Based on Kernel Combination." *Neurocomputing*, SI: Recent Advances in SVM, 211 (October 26, 2016): 98–105. https://doi.org/10.1016/j.neucom.2015.10.139.

University, Stanford. "Inventory Indicates Who Goes Solar and Why." Stanford News, December 19, 2018. https://news.stanford.edu/2018/12/19/inventory-indicates-goes-solar/.

Yuan, Jiangye, Hsiu-Han Lexie Yang, Olufemi A. Omitaomu, and Budhendra L. Bhaduri. "Large-Scale Solar Panel Mapping from Aerial Images Using Deep Convolutional Networks." In *2016 IEEE International Conference on Big Data (Big Data)*, 2703–8. Washington DC,USA: IEEE, 2016. https://doi.org/10.1109/BigData.2016.7840915.

Yue, Songqing. "Imbalanced Malware Images Classification: A CNN Based Approach." *ArXiv:1708.08042 [Cs, Stat]*, August 26, 2017. http://arxiv.org/abs/1708.08042.