

# Image Processing: Seeing the World Through the Eyes of SAS® Viya™

Leigh Ann Herhold and Ivan Gomez, Zencos Consulting LLC

## ABSTRACT

Ever wonder how the algorithms used by Facebook and Google detect your friends in your photos? Image recognition and classification algorithms, such as deep neural networks, can extract important information from photos and classify them almost instantly after you post a picture. Not only is this process useful on social media, but there are numerous applications of image classification algorithms in healthcare, manufacturing, and security screenings. Using the latest machine learning capabilities available in SAS® Viya® for text and image processing, organizations can leverage in-memory processing with SAS® Cloud Analytic Services (CAS) and enhanced parameter tuning to develop more sophisticated deep learning models. In this paper, we discuss key components of building an image classification algorithm.

## INTRODUCTION

Over the past two years, artificial intelligence has gained significant interest across many industries, ranging from self-driving cars, airport security, virtual assistants with speech capabilities, to enhanced medical imaging processing. Social network platforms immediately offer recommendations to tag your friends after uploading a new picture, as well as the ability to search photo libraries for objects and activities. Want to see all your pictures from your snowboarding trips? Easy. Just search for “snowboarding” and let the application take care of the rest.

However, it is less known that all these developments share common roots and have become possible due to the recent explosion in interest in a subset of machine learning techniques called deep learning or deep neural networks. Figure 1 shows the Interest over time<sup>1</sup> for Google searches related to Deep Learning. This graph shows that since 2014, deep learning has received more and more interest leading to all time high interest levels in 2018.

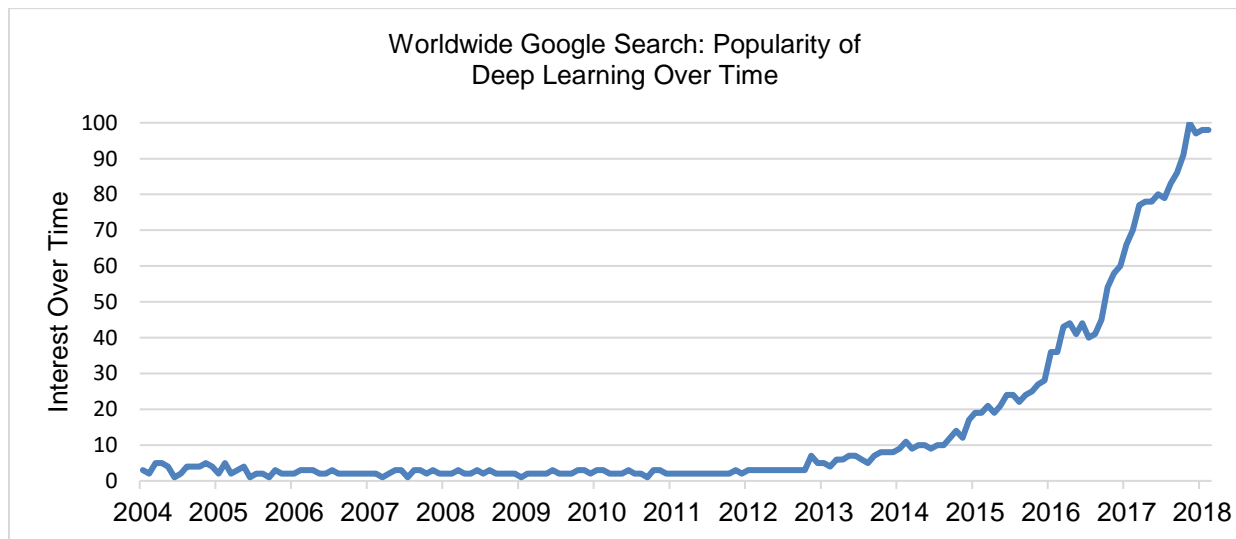


Figure 1. Worldwide Google Search Interest of Deep Learning over time

<sup>1</sup> Interest over time: As defined by Google, Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term.

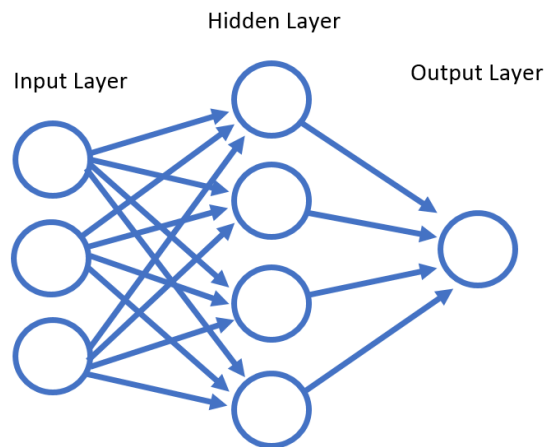
All these recent developments and trends may mask the fact that even though neural networks have quickly become the focus of many machine learning applications, they are not exactly a new concept. Many of the essential components and mathematical designs that power today's neural networks were originally proposed in the 1940s, while key algorithmic advances were made in the 1970s (McCulloch 1943, Werbos 1974). However, at that time these ideas seemed impossible and their implementation was not feasible because of the limited computational power available and lack of large datasets necessary to train the models. Both limitations have been overcome in recent years through technological developments in computing power, cloud computing, parallel and GPU processing, as well as vast amounts of data collected and preprocessed over decades. This, combined with advances in machine learning algorithms has enabled researchers to iterate more quickly over ideas to improve the state of the industry. At the same time, organizations now are able to identify opportunities to leverage these techniques (Ng 2017).

Version 3.3 of SAS Viya introduces the SAS Deep Learning toolkit, enabling users to leverage the Cloud Analytics Services in-memory engine to easily create, train, and tune deep fully-connected neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) for a variety of applications including text mining, facial recognition, and object detection.

This paper provides a high-level overview of neural networks, discusses modern-day applications of image analytics, and walks through a basic facial recognition model using Python and SAS Viya.

## INTRODUCTION TO NEURAL NETWORKS

Neural networks are analytical models within the machine learning field that were initially derived to mimic how neurons in the brain function. Although there are a variety of neural networks, all neural networks are composed of a few core components including an input layer, hidden layer(s), and an output layer. Each layer is comprised of nodes. Figure 2 shows a diagram of the core components of a neural network with a single hidden layer.



**Figure 2. Neural Network Diagram**

Neural networks are commonly named after their depth, or number of layers. Oftentimes the *depth* of a network will impact the predictability and accuracy of the model. A deep neural network is one that has many hidden layers between the data input and output layers. Hence, deep learning is a term used to describe larger and more complicated networks that are used for extracting valuable information from unstructured data such as video or images.

As neural networks deepen, and hidden layers are added, more complex features can be detected within the data, further improving the output predictions of the model. One advantage of neural networks compared to other predictive models is that they can model complex, non-linear relationships for better classification. In the context of facial recognition, shallower sections of a network may be able to detect lines whereas deeper layers can detect more complex features such as parts of faces.

## PRACTICAL APPLICATIONS OF IMAGE ANALYTICS

Image analytics is applicable to a wide variety of industries and continues to change the consumer experience, drive down cost, provide greater opportunity for effective marketing, and increase security. Society is already reaping the benefits of using these algorithms.

### SOCIAL MEDIA AND COMMUNICATION

Social media is one of the most common and well-known spaces in which facial recognition is used and recognized. Approximately 1.1 trillion photos were shared globally in 2016, and that number continues to rise (Perret 2017). Social media companies and applications including Instagram and Facebook contribute to photo sharing, and both Facebook and Google have implemented effective facial recognition algorithms to detect and label people within photos.

### MEDICINE: MEDICAL IMAGING

The medical field also has a plethora of opportunities for machine learning models for object detection and recognition. Medical imagery including MRIs, CAT scans, and X-rays are constantly used to provide medical evaluations and care to patients. One application of machine learning within the medical field is using CNN models to quickly and accurately detect cancer by comparing cancerous images to non-cancerous images. Proper image detection could even help pave the way for effective telemedicine diagnoses over video calls, providing more accessible care to remote patients. In addition, CNNs can be used in both 2D and 1D and can be used to detect differences in heartbeats and better assess heart health (Nu 2017, “Neural Style Transfer: 1D and 3D Generalizations”).

### INSURANCE

There are numerous opportunities for machine learning algorithms for automotive and insurance companies. Insurance companies such as Mitchell are working towards using machine learning to assess vehicle damage “so that vehicle damage can be assessed within seconds and consistent decisions made” (Huetter 2016). Uploading images and obtaining a quick damage assessment will not only improve the customer experience but also improve efficiency in the claims process.

### FINANCE

The financial industry is continually plagued with a variety of pervasive types of fraud. Fraud losses continue to increase into the billions with an estimated total of \$181 billion in losses from 2016. Wire fraud, credit card fraud, identity theft, and check fraud are among the top ten types of fraud in 2016 (McKenna 2017). With advances in image analytics and machine learning, financial corporations including banks, fintechs, money services businesses, and casinos can use image analytics for check verification, extracting information from IDs or scanned documents, and assessing customer risk with real-time facial recognition to mitigate fraud losses.

## BUILDING A FACE RECOGNITION MODEL USING PYTHON AND SAS VIYA

For this paper, we developed a convolutional neural network (CNN) for facial recognition leveraging the following:

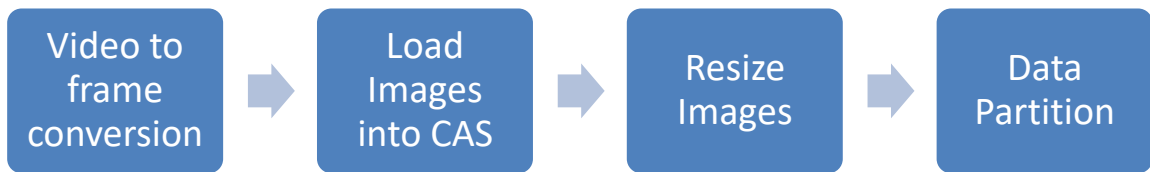
- **The RestNet-50 CNN architecture (He 2015):** As summarized in the SAS Deep Learning Programming Guide, “ResNet is a residual network developed by Kaiming He et al. ResNet was the winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015. ResNet features special skip connections and a strong use of batch normalization. The architecture also notable because it does not have any fully connected layers at the end of the network. ResNets are considered by many (in 2016) to be state of the art convolutional neural network models.”
- **The RestNet-50 model parameter file available in the SAS Deep Learning Models and Tools:** provides parameter weights of the original RestNet-50 model which won multiple computer vision competitions.

- **Facial data:** approximately 25 seconds of video at 30 frames per second in .MOV format, capturing several facial expressions for each of six different Zencos consultants. The video was decomposed into individual frames that were later used to train the model with the objective of recognizing each consultant.
- **Python:** programming language to interface with CAS through the SAS SWAT package.

Data modeling tasks normally require data cleaning and transformation to convert raw data into the proper formatting for modeling. For image analytics, these steps often involve converting video files into images, labeling those images, and resizing them to match the model requirements. Once data is ready for modeling, it can be analyzed with machine learning algorithms to detect and learn the patterns within it. The next two sections of this paper detail the data preprocessing and model training process that enabled our model to learn the facial features to ultimately identify Zencos consultants.

## IMAGE DATA PREPARATION IN SAS VIYA

Preparing the facial data for modeling consisted of a four-step process shown in Figure 3:



**Figure 3: Data Preparation Steps**

### VIDEO TO FRAME CONVERSION

Each video was converted to individual pictures using the `video_to_frames` function defined below:

```

def video_to_frames(video, path_output_dir):
    # count is the frame index
    vidcap = cv2.VideoCapture(video)
    count = 0
    while vidcap.isOpened():
        success, image = vidcap.read()
        if success:
            cv2.imwrite(os.path.join(path_output_dir, '%d.jpg') % count,
                        image)
            count += 1
        else:
            break
    vidcap.release()
  
```

This function accepts both the path to the video files and the path to the output directory to store the images. Once executed, it decomposes the video into individual .jpg files. Executing this conversion on all six videos generated 4,544 images that were stored in separate directory for each individual.

### LOAD IMAGES INTO CAS

The `loadImages` action from the Image CAS Action Set provided the functionality to load the images into an in-memory Cloud Analytic Services (CAS) table. The following snippet loaded 4,544 images into a CAS table called `Zencos_Faces`:

```

s.loadImages(casout={"name":"Zencos_Faces", "replace":True},
             path=zencos_faces_path,
             labelLevels=-1,
             recurse=1)
  
```

In this case, a combination of the `labelLevels` and `recurse` options were leveraged to automatically label all images as shown in Figure 4. This can be particularly useful when analyzing images with many categories.

	<code>_image_</code>	<code>_size_</code>	<code>_path_</code>	<code>_label_</code>	<code>_type_</code>	<code>_id_</code>
0	<code>b"\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00...</code>	429610	<code>/data/Zencos_Faces/Ivan/149.jpg</code>	Ivan	jpg	150
1	<code>b"\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00...</code>	368184	<code>/data/Zencos_Faces/Leigh_Ann/376.jpg</code>	Leigh_Ann	jpg	1000
2	<code>b"\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00...</code>	400509	<code>/data/Zencos_Faces/Chris/432.jpg</code>	Chris	jpg	2000
3	<code>b"\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00...</code>	404764	<code>/data/Zencos_Faces/Tanyss/60.jpg</code>	Tanyss	jpg	3000

Figure 4. CAS Table with Loaded and Labeled Images

## RESIZE IMAGES

Input images to the ResNet-50 model needed to be resized to match the size expected by the model. The `processImages` action from the Image Action Set enables performing several core image transformations, including resizing. A new table, `Zencos_Faces_resized`, contains the input images with a new size of 224 by 224 pixels:

```
zencos_faces = s.CASTable('Zencos_Faces')
zencos_faces.processimages(casout={'name':'Zencos_Faces_resized',
                                  'replace':True},
                           imagefunctions=[{'functionoptions':{'width':224,
                                                                'functiontype':'RESIZE',
                                                                'height':224}}]
                           )
```

Figure 5 below shows an example of four different images from the `Zencos_Faces_resized` Table.

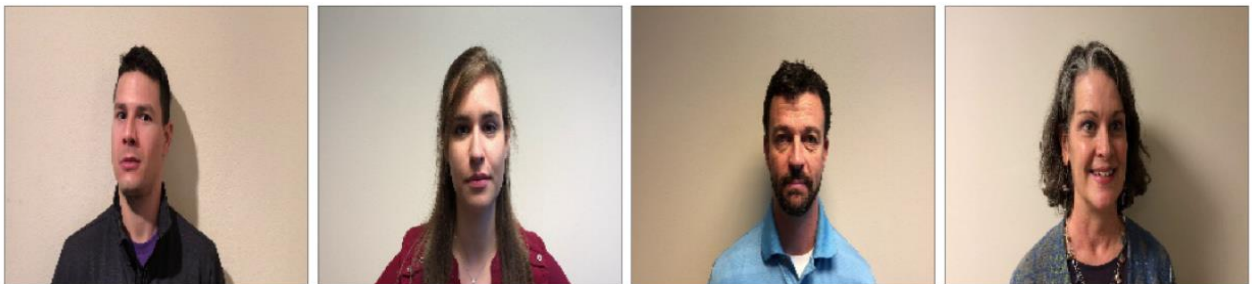


Figure 5. Examples of Resized Facial Images

## PARTITION

The last step before training the ResNet-50 model was to partition the input data to use 80% of the images for training and the remaining 20% for validation:

```
s.stratified(table={"name":"Zencos_Faces_Resized",
                   "groupBy":{"_label_"}
               },
             display={"names":"STRAFreq"},
             output={"casOut":{"name":"Zencos_Faces_Partitioned",
                              "replace":True},
                   "copyVars":"ALL"},
             sampct=80,
             sampct2=20,
             partind=True,
```

```

seed=10,
outputTables={"names":{"STRAFreq"},
              "replace":True}
)

```

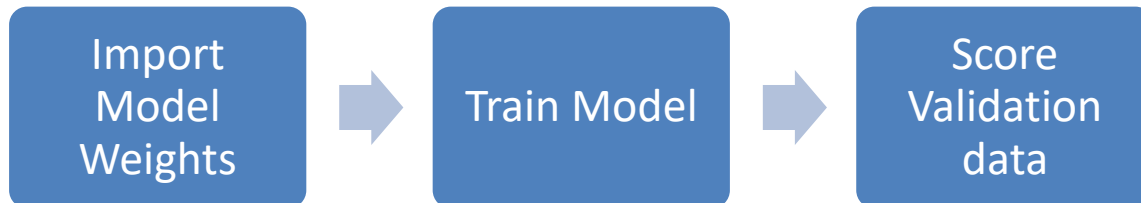
The `Zencos_Faces_Partitioned` CAS table represents the input table used to train the Resnet-50 model. A breakdown of the training and validation split is shown in Table 1:

Group ID	Label	Total number of images	Training	Validation
0	Chris	775	620	155
1	Chris_StJeor	774	619	155
2	David	765	612	153
3	Ivan	623	498	125
4	Leigh_Ann	776	621	155
5	Tanyss	831	665	166
	<b>Total</b>	<b>4544</b>	<b>3635</b>	<b>909</b>

**Table 1. Breakdown of input data for training and validation**

## MODEL TRAINING THROUGH TRANSFER LEARNING

The training process for the ResNet-50 model can be summarized in three steps shown in Figure 6:



**Figure 6. Model Training Process**

### IMPORT MODEL WEIGHTS

Deep learning models are, in essence, very complex mathematical functions with millions of parameters that need to be calibrated to ultimately generate accurate predictions. This calibration process is what is commonly known as model training.

Training an image classification model from scratch normally requires millions of input images and hundreds of hours of computational runtime. However, the SAS Deep Learning toolkit enables data scientists to leverage a technique called transfer learning to greatly decrease the input data size requirements and the total amount of times it takes to train any of these models.

Instead of starting from scratch, transfer learning works by initializing a model with parameters that have been trained for weeks on very large datasets. Essentially, the knowledge these parameters learned from the original data can be transferred to a new model and then tuned to learn the nuances of the new data (Ng 2017).

In this example, the ResNet-50 contains over 23 million parameters that need to be trained. Instead of starting from random initialization, this model was initialized with the parameters from the original ResNet-

50 model trained on the ImageNet <sup>2</sup> dataset using the `dlImportModelWeights` CAS action (Russakovsky 2015):

```
s.dlimportmodelweights(model=modelName,
                      modelWeights={'caslib':'CASUSER',
                                    'name':'tkdlWeights',
                                    'replace':1},
                      formatType="CAFFE",
                      weightFilePath=modelFile)
```

In this snippet, `modelName` is the CAS table that contains the definition of the RestNet-50 model, `modelFile` is the path to the model parameter file downloaded from the SAS Deep Learning Models and Tools site, and `tkdlWeights` represents the CAS table where the imported weights are stored for later use.

## TRAIN MODEL

Having created the input dataset and initialized the model weights, the model was trained on new data using the `dlTrain` action:

```
s.deepLearn.dlTrain(inputs=[{"name": "_image_"}],
                    modelTable={"name": modelName},
                    modelWeights={"name": "{}_weights".format(modelName),
                                  "replace": True},
                    bestWeights={"name": "{}_best".format(modelName),
                                 "replace": True},
                    table={"name": "Zencos_Faces_Partitioned",
                           "where": "_PartInd_ = 1"},
                    initWeights={"name": "tkdlWeights",
                                  "where": "_layerid_ < 125"},
                    optimizer={"algorithm": {"method": "ADAM",
                                             "useLocking": True}},
                             "maxEpochs": 6},
                    target="_label_"
                    )
```

The key arguments to this action are:

- `inputs` – The variable that contains the image data in the input dataset.
- `modelTable` – CAS table that contains the definition of the RestNet-50 model.
- `modelWeights` and `bestWeights` – Output CAS tables that store the resulting model parameters.
- `table` – Input table created during the data preparation process.
- `initWeights` – CAS table containing the model parameters imported in the previous step.
- `target` – The target variable to predict.

By using this action, SAS Viya provides a summary of the model created, including number of layers, types of layers, and number of parameters shown in Figure 7:

---

<sup>2</sup> ImageNet Dataset: This dataset contains over 14 million images classified in 100 categories and has become a benchmark for computer vision tasks.

## § ModelInfo

	Descr	Value
0	Model Name	resnet50
1	Model Type	Convolutional Neural Network
2	Number of Layers	126
3	Number of Input Layers	1
4	Number of Output Layers	1
5	Number of Convolutional Layers	53
6	Number of Pooling Layers	2
7	Number of Fully Connected Layers	0
8	Number of Batch Normalization Layers	53
9	Number of Residual Layers	32
10	Number of Weight Parameters	23467200
11	Number of Bias Parameters	53190
12	Total Number of Model Parameters	23520390
13	Approximate Memory Cost for Training (MB)	5481

Figure 7. Model Information Provided by the dlTrain CAS Action

## SCORE VALIDATION DATA

The trained model was used to score the validation partition of the images. The `dlscore` action was used to score the holdout sample:

```
s.dlscore(model=modelName,  
          initWeights={'caslib':'CASUSER',  
                      'name':"{}_weights".format(modelName)},  
          table={'caslib':'CASUSER',  
                "name":"Zencos_Faces_Partitioned",  
                "where":"_PartInd_=2",  
                },  
          copyvars=['_label_'],  
          topProbs = 5,  
          miniBatchSize = 3,  
          casout={'caslib':'CASUSER',  
                 'name':'ScoredValidation',  
                 'replace':True})
```

The results of this action show the model accuracy statistics with respect to misclassification error. This model achieved 100% accuracy as shown in Figure 8:



## § ScoreInfo

	Descr	Value
0	Number of Observations Read	909
1	Number of Observations Used	909
2	Misclassification Error (%)	0
3	Top 5 Misclassification Error (%)	0
4	Loss Error	0.000029

**Figure 8. Model Accuracy Statistics Provided by dIScore CAS Action**

It is important to note that while achieving 100% accuracy with this validation partition is a positive indicator that the model has learned features that differentiate each person, this may not be the final step in creating a model.

In this example, all the images have similar backgrounds, the same resolution, and contain a single face, facilitating model training and making it easier for the model to tell people apart. Common examples of additional challenges that arise when analyzing pictures with multiple people in the frames include finding and cropping individual faces, adjusting for different facial positions and angles, accounting for differences in lighting conditions, etc. In those cases, one practical approach would be to develop a pipeline of deep learning models where each model solves one challenge and provides inputs to the next model.

## CONCLUSION

Advances in computing power and availability of data have enabled exponential growth of use cases for machine learning and artificial intelligence applications that are reshaping today's businesses and setting trends for the future, especially within the realm of computer vision and image analytics.

Social media platforms have an ever-growing user base constantly uploading pictures and videos of everyday life activities. By analyzing and applying different types of image analytics for object identification and detection, facial recognition, and activity tracking, these organizations can better understand their customer base and use that information to develop features relevant to their customers' lives, help users connect with each other based on common interests, increase revenues through targeted marketing, etc.

In addition to social media, virtually every industry and company can benefit from the implementation of deep learning models to streamline current operations and open the doors to new business opportunities. Image analytics has the potential to change the insurance industry and reduce the claim processing times with models to assess damage. Deep learning models can help bring medical care to remote corners of the world through telemedicine. Facial recognition models and biometric scans are being used to improve airport security and unlock personal devices.

Developing models that can accomplish these tasks requires modern analytics platforms capable of processing vast amounts of data in reasonable amounts of time. The SAS Deep Learning toolkit enables users to leverage the Cloud Analytics Services for massive parallel in-memory processing. Furthermore, this toolset integrates with open source pre-trained models and enables rapid creation, training, and customization of deep learning model pipelines including fully-connected neural networks, convolutional neural networks, and recurrent neural networks for a variety of applications including text mining, facial recognition, and object detection.

## REFERENCES

- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *ArXiv preprint arXiv:1512.03385*. doi:10.1109/cvpr.2016.90
- Huetter, J. (2016, October 11). "Tractable photo-analyzing AI to assess vehicle damage for Mitchell." Accessed February 24, 2018. Retrieved from <http://www.repairerdrivenews.com/2016/10/11/tractable-ceo-says-ai-will-assess-vehicle-damage-for-mitchell-company-specializes-in-ai-photo-analysis/>
- McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*. 5 (4): 115–133. Accessed by <http://www.cse.chalmers.se/~coquand/AUTOMATA/mcp.pdf>
- McKenna, F. (2017, January 17). "Top 10 Fraud Types for 2017 Based on Losses." Accessed February 24, 2018. Retrieved from <http://frankonfraud.com/fraud-reporting/top-10-fraud-losses-for-2016-and-where-they-are-headed-now/>
- Ng, A. 2017. "Intro to Deep Learning: Why is Deep Learning taking off?". [Video file]. Coursera. Accessed February 24, 2018. Retrieved from <https://www.coursera.org/learn/neural-networks-deep-learning/lecture/pragm/why-is-deep-learning-taking-off>
- Ng, A. 2017. "Neural Style Transfer: 1D and 3D Generalizations." [Video file]. Coursera. Accessed February 24, 2018. Retrieved from <https://www.coursera.org/learn/convolutional-neural-networks/lecture/OkHJ3/1d-and-3d-generalizations>
- Ng, A. 2017. "Practical advices for using ConvNets: Transfer Learning." [Video file]. Coursera. Accessed February 24, 2018. Retrieved from <https://www.coursera.org/learn/convolutional-neural-networks/lecture/4THzO/transfer-learning>
- Perret, E. (2017, January 29). "Here's How Many Digital Photos Will Be Taken in 2017." Accessed February 24, 2018. Retrieved from <https://mylio.com/true-stories/tech-today/how-many-digital-photos-will-be-taken-2017-repost>
- Russakovsky\*, O; Deng\*, J; Su, H.; Krause, J; Satheesh, S; Ma, S; Huang, Z; Karpathy, A.; Khosla, A; Bernstein, M; Berg, A; Li Fei-Fei, L. (\* = equal contribution) "ImageNet Large Scale Visual Recognition Challenge." *IJCV*, 2015.
- SAS Institute Inc. 2017. SAS® Visual Data Mining and Machine Learning 8.2: Deep Learning Programming Guide. Cary, NC: SAS Institute Inc. Accessed February 24, 2018. Retrieved from <http://documentation.sas.com/?docsetId=casdlpg&docsetTarget=titlepage.htm&docsetVersion=8.2&locale=en>
- Werbos, P. (1974). *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Harvard University.

## RECOMMENDED READING

- Deep Residual Learning for Image Recognition by Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (<https://arxiv.org/abs/1512.03385>)
- SAS® Visual Data Mining and Machine Learning 8.2: Deep Learning Programming Guide (<http://documentation.sas.com/?docsetId=casdlpg&docsetTarget=titlepage.htm&docsetVersion=8.2&locale=en>)
- Neural Networks and Deep Learning by deeplearning.ai
- Visualization of the ResNet-50 network structure (<http://ethereon.github.io/netscope/#/gist/db945b393d40bfa26006>)

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Leigh Ann Herhold  
Zencos Consulting  
(919) 237-9079  
[laherhold@zencos.com](mailto:laherhold@zencos.com)

Ivan Gomez  
Zencos Consulting  
(919) 576-0614  
[igomez@zencos.com](mailto:igomez@zencos.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.