

Applying Image Recognition to Insurance

0.00	0.00	0.00	0.00	0	MAR 12,000	0.75	1.00	0.00	0.00	0.00	0.00	0	30
0.00	0.00	0.00	0.00	0	MAR 13,000	1.05	1.30	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	MAR 14,000	1.50	1.75	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	MAR 15,000	2.05	2.30	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	MAR 16,000	2.60	2.85	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	MAR 17,000	3.25	3.50	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	MAR 18,000	3.85	4.10	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	OCT 15,000	0.10	0.35	0.00	0.00	0.00	0.00	0	35
0.00	0.00	0.00	0.00	0	OCT 16,000	0.30	0.55	0.00	0.00	0.00	0.00	0	110
0.00	0.00	0.00	0.00	0	OCT 17,000	0.85	1.10	0.00	0.00	0.00	0.00	0	25
0.00	0.00	0.00	0.00	0	OCT 18,000	1.65	1.90	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	NOV 15,000	2.50	2.75	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	NOV 16,000	0.85	1.00	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	NOV 17,000	1.45	1.65	0.00	0.00	0.00	0.00	0	10
0.00	0.00	0.00	0.00	0	NOV 18,000	2.10	2.35	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	NOV 19,000	2.86	3.10	0.00	0.00	0.00	0.00	0	0
0.00	0.00	0.00	0.00	0	DEC 12,000	3.70	3.95	0.00	0.00	0.00	0.00	0	0
3.10	0.00	0.00	0.00	0	DEC 13,000	0.30	0.55	0.00	0.00	0.00	0.00	0	0
2.55	0.00	0.00	0.00	0	DEC 14,000	0.50	0.75	0.00	0.00	0.00	0.00	0	10
2.00	0.00	0.00	0.00	0	DEC 15,000	0.85	1.10	0.00	0.00	0.00	0.00	0	20
1.60	0.00	0.00	0.00	0	DEC 16,000	1.25	1.50	0.00	0.00	0.00	0.00	0	10
1.30	0.00	0.00	0.00	0	DEC 17,000	1.85	2.10	0.00	0.00	0.00	0.00	0	0
0.95	0.00	0.00	0.00	0	DEC 18,000	2.50	2.75	0.00	0.00	0.00	0.00	0	20
4.40	0.00	0.00	0.00	0	MAR 12,000	3.20	3.45	3.30	0.00	3.30	3.30	6	6
3.75	0.00	0.00	0.00	0	MAR 13,000	0.75	1.00	0.00	0.00	0.00	0.00	0	30
3.30	0.00	0.00	0.00	0	MAR 14,000	1.05	1.30	0.00	0.00	0.00	0.00	0	0
2.85	0.00	0.00	0.00	0	MAR 15,000	1.50	1.75	0.00	0.00	0.00	0.00	0	0
2.45	0.00	0.00	0.00	0	MAR 16,000	2.05	2.30	0.00	0.00	0.00	0.00	0	20
2.05	0.00	0.00	0.00	0	MAR 17,000	2.60	2.85	0.00	0.00	0.00	0.00	0	30
1.75	0.00	0.00	0.00	0	MAR 18,000	3.25	3.50	0.00	0.00	0.00	0.00	0	40
1.60	0.00	0.00	0.00	0	OCT 15,000	0.85	1.10	0.00	0.00	0.00	0.00	0	10
0.90	0.00	0.00	0.00	0	OCT 16,000	0.10	0.35	0.00	0.00	0.00	0.00	0	35
0.45	0.00	0.00	0.00	0	OCT 17,000	0.30	0.55	0.00	0.00	0.00	0.00	0	110
0.30	0.00	0.00	0.00	0	OCT 18,000	0.85	1.10	0.00	0.00	0.00	0.00	0	25
0.25	0.00	0.00	0.00	0	NOV 15,000	1.65	1.90	0.00	0.00	0.00	0.00	0	70
1.45	0.00	0.00	0.00	0	NOV 16,000	2.50	2.75	0.00	0.00	0.00	0.00	0	0
					NOV 17,000	0.85	1.00	0.00	0.00	0.00	0.00	0	0



Applying Image Recognition to Insurance

AUTHOR

Kailan Shang, FSA, CFA, PRM, SCJP

SPONSOR

Society of Actuaries Research
Expanding Boundaries Pool

Caveat and Disclaimer

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries or its members. The Society of Actuaries makes no representation or warranty to the accuracy of the information.

Copyright © 2018 by the Society of Actuaries. All rights reserved.

CONTENTS

Acknowledgments	4
Executive Summary	5
Section 1: Introduction	6
Section 2: Insurance Application	7
Section 3: Challenges	12
Section 4: Data Processing	14
4.1 Data Transformation	14
4.2 Data Augmentation	16
4.3 Feature Extraction	16
4.4 Autoencoders	17
Section 5: Model	18
5.1 Fully Connected Neural Network.....	19
5.2 Convolutional Neural Network	20
5.3 Activation Function	21
5.4 Pooling	23
5.5 Normalization	23
5.6 Regularization	24
5.7 Calibration.....	25
Section 6: Example: Driver Behavior Assessment Using Image Recognition	28
6.1 Data Processing	28
6.2 Model	29
6.3 Training	29
6.4 Validation	32
6.5 Application	32
Section 7: Conclusion	33
References	34
About The Society of Actuaries	35

Acknowledgments

The author would like to thank all members of the Project Oversight Group (POG) tasked with providing governance on this research project. The POG provided insightful inputs, especially on how to improve the report's relevance to actuarial works.

The Effective ERM Stakeholder Engagement POG members are as follows:

- Jeff Plank
- Ronora Stryker
- Gwen Weng

The author thanks Julia Jiang and Joanna Shang for their input on the application of image recognition techniques, as well as Jan Schuh for managing the research activities and the sponsorship and funding support of the Society of Actuaries.

Executive Summary

Image recognition techniques are used in many areas of modern life, such as facial recognition, image searches, and optical character recognition. Although image recognition models are very complicated, the process from image data to prediction result is no different from that of traditional statistical modeling. Image recognition models use stacked traditional statistical models to approximate the complicated relationship between input and output that is unlikely to be represented by a single function. Most methods of normalization and regularization used in image recognition are also used in traditional models. The tools available for developing image recognition make the use of such models possible for actuaries with a background in statistics.

Automated image recognition may be applied to the insurance industry in many areas. It can be used to improve customer service, such as using facial recognition for identity verification and faster underwriting and claim processing. With satellite images, precise agricultural insurance pricing and risk assessment can be achieved with crop, weather and landscape information for areas as small as a few kilometers. Home insurance pricing may be improved as well using traditionally difficult-to-capture images of the insured property such as the roof. Extreme risk events such as floods, hurricanes, tornadoes and wildfires can be monitored and the impact on insurance claims updated in real time with the aid of automated image recognition. Medical images can provide richer information for health insurance underwriting and pricing. Optical character recognition can help digitize documents and facilitate information saving, searching and sharing.

There is no doubt about the benefit of using image recognition techniques in the long run. Before applying them in practice, however, a few factors need to be considered. The amount of available data needs to be big, and the cost of data collection needs to be reasonable. The accuracy level of the model should be acceptably high, and the adverse impact of a wrong prediction should be manageable. The improvement of decision making needs to have sufficient financial benefits to offset the large investment in the techniques. As with the adoption of any new technology, it is a cost-benefit analysis to consider the investment of resources versus the gain from the application.

Technical challenges also exist for a successful application of automated image recognition. Existing models are usually trained to identify the objects in an image, but for insurance applications, more useful information may be the behavior of the object. For example, is a driver using a cell phone while driving? This often leads to more customized model training using relevant image data. Data collection and model training may take a long time and require many resources. The current accuracy level of most advanced models is between 70% and 90%. A small error could lead to customer complaints and potential reputational risk. Cyber risk may also increase when automated image recognition techniques are used. For example, if facial recognition is used to confirm identification, the program may be hacked so that it accepts illegal requests and allows illegal access to private data.

Further developments are needed to improve model accuracy so the models are intelligent enough to improve users' analysis and decision making. At the same time, image recognition provides many opportunities for actuaries. Equipped with both the industry knowledge and the technical skills, actuaries can help link image recognition with risk assessment and decision making in a meaningful way. They can help design image recognition model structures that can solve more complicated insurance-related issues and help validate image recognition conclusions using existing models based on alternative data sources.

This report includes an example of how image recognition models can be used to improve an insurance-related issue: predicting driver behavior based on images captured during driving. Does the driver text or call while driving? Does the driver focus on the road while driving? The example covers data processing, model building, model validation and prediction. The source code is also made public on GitHub for educational purposes (<https://github.com/windwill/imgRecognition>). Hopefully this will shorten the learning curve for people with statistical backgrounds and encourage more applications of image recognition in the insurance world.

Section 1: Introduction

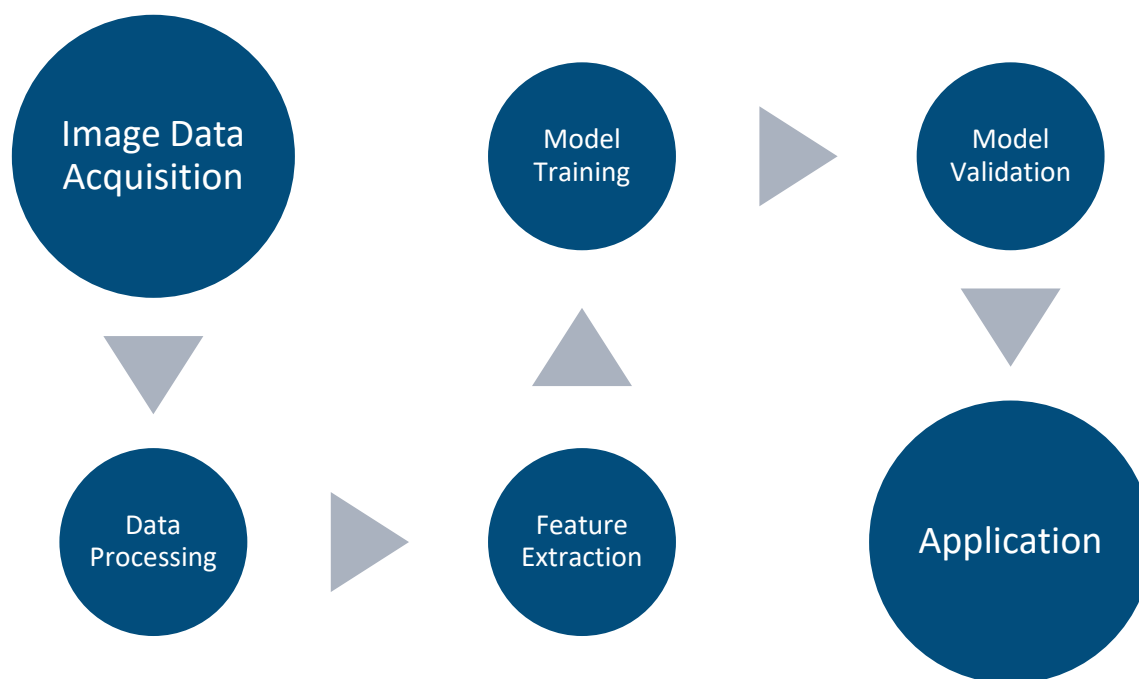
With advances in computer vision, the Internet of Things (IoT) and artificial intelligence, many industries are experiencing the changes and challenges brought by new technologies such as image recognition. Image recognition is the process of using a machine or computer to detect an object, a feature or other useful information from an image or a sequence of images such as a video. A common example of image recognition is optical character recognition (OCR). Without human reading and recording, a scanner can convert an image of texts to a text file by identifying the standard characters in the image. OCR may also be applied to recognize a license plate in a photo. It can be used for automatic red light cameras, speed cameras or parking meters. Facial recognition, automatic inspection, cancer prognosis, species identification and automated driving are other examples of image recognition applications.

Image recognition is expected to affect many areas of the insurance industry. It can improve interactions with customers by providing more automation and higher efficiency. Image data is another dimension of data that can be collected and analyzed to improve decision making. The insurance industry is gradually applying image recognition technology. Areas in which automated image recognition is used include customer identity confirmation, underwriting, claim analysis, disaster monitoring and document digitization. The changes improve the vitality of the insurance business and have a profound impact on customer behaviors, risk assessment and information sharing. For example, with facial recognition, it will be more convenient for customers to apply for insurance coverage or claim insurance benefits. Images of an insured property can be used for underwriting. Using drones to take photos of house roofs can improve home insurance underwriting through image recognition. Images of drivers may also be used to detect unsafe driving behaviors, which can help risk assessment and improve driving safety. Images of an insurance accident or a natural disaster can help insurance companies quickly identify issues, allocate resources and estimate losses.

A typical image recognition application is complicated but the process is similar to traditional statistical analysis. Figure 1 shows a high-level example of this process. Images are represented as data and then processed. Common processing includes data compression, noise reduction and contrast enhancement to provide better model input. Features such as lines, intersections, borders, shapes and characters may be extracted from images, or they can be learned automatically from the raw data by deep learning models such as a convolutional neural network. Models are trained to link image data with interested outcomes such as the object type and behaviors. Models are then validated to make sure they are powerful enough to help predict the outcome. If a model generates satisfactory results, it will then be used for real business application.

Figure 1

Sample Image Recognition Process



This paper explores the current status of image recognition technology, models for image recognition, and various possible applications of image recognition to insurance. By leveraging on existing actuarial education, it tries to fill the knowledge gap between traditional statistical and image recognition models and encourage more actuarial pioneering work in this field. The report proceeds as follows:

- Section 2, “Insurance Application,” discusses how image recognition can be used for insurance applications and the possible impacts it may have.
- Section 3, “Challenges,” examines the potential difficulties of using image recognition in the insurance industry, including model fine-tuning, model accuracy, and fraud detection. It also touches on the areas where actuarial knowledge is helpful in meeting these challenges.
- Section 4, “Data Processing,” looks at how image data can be processed before being used for recognition tasks.
- Section 5, “Model,” introduces popular image recognition models and their links to actuarial models.
- Section 6, “Example: Driver Behavior Assessment Using Image Recognition,” presents an example of using image recognition to assess driving behavior. It includes the details of the entire process from data processing and modeling to result validation and application. It exemplifies the knowledge, tools and efforts that are needed for an image recognition project.
- Section 7, “Conclusion,” summarizes the key points of this research and concludes the main body of the report.

Section 2: Insurance Application

How can automated image recognition be used by the insurance industry? To answer this question, we need to understand why we need automated image recognition. The first reason is that images can provide additional

information that traditional data sources cannot. For example, if we see an image of a driver using a cell phone while driving, it tells us that the driver is more likely to be a high-risk client. The second reason is that automated image recognition can speed up certain tasks and reduce service waiting time. But why should we rely on an automated system instead of human judgment? From a resource perspective, there may be too many images for a human being to process in an efficient and timely manner. From a cost perspective, it may be cheaper to use models rather than humans to retrieve information from images.

Image recognition techniques are already prevalent in many industries. Identity authentication, automatic photo tag suggestion, image search, and product image matching are widely used in social media services, online shopping and mobile services. Existing applications are mostly related to improving customer service and security. For example, facial recognition is the most popular application of image recognition in the banking industry. Many banks allow users to log in to mobile banking services using automatic facial recognition on their smartphones. Other applications of image recognition in banking include the following:

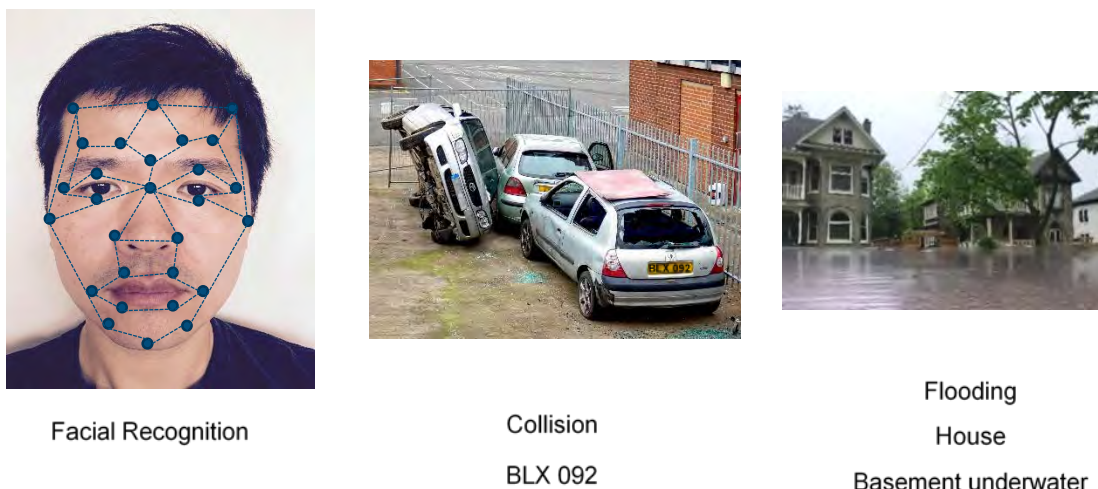
1. **Fraud detection.** Facial recognition can reduce the fraudulent use of data and banking services. For example, facial recognition may be required for online banking, identity authentication at ATMs and employee log-in to the system. It may also prevent customers from registering for services using different identities. Real-time alerts enabled by automatic recognition techniques provide instant knowledge of risks and allow timely actions to minimize loss.
2. **Verification of signatures or other handwriting.** By automatically assessing the similarity of a signature to the one on record, a customer's identity can be verified to approve banking transactions. For example, a check can be deposited by taking a photo of it.
3. **Video surveillance.** To improve the security of bank facilities, image recognition programs can be integrated with real-time camera streaming to prevent unauthorized access and identify suspects after an event. This can save hours of manual searching through traditional surveillance footage. Banks may also analyze videos of different branches to understand the customer's path from entrance to exit. Automatic image recognition can help collect important information on wait time, the customer service process, and locations where the customer experience can be improved and advertisements optimized.
4. **Personalized service.** Facial biometrics can be used to identify important clients and provide personalized services to improve customer satisfaction.
5. **Interactive marketing.** Based on potential customers' image data on social networks, marketing and sales efforts can be tailored for better success. Customers may be classified into different types with corresponding marketing strategy and product solutions.

Like other fields, the insurance industry can benefit from automated image recognition in many areas. Some applications can leverage mature techniques, whereas others need customization and improvement.

1. Image recognition can be used to improve customer services. This is similar to what is happening in the banking industry. For example, if facial recognition—illustrated in Figure 2—can be used to verify identity, it is easier for existing customers to review their policy information and get services. Facial recognition is also more secure than user names and passwords. For future customers, applying for insurance can be much easier with image recognition. Customers can take a photo of their required documents and e-mail them to the insurance company. The image recognition engine translates the images into the required information, and the application may be approved in just a few seconds if everything goes well. The same advantage can apply to the claim process. For instance, an auto insurance claim may be made simply by taking a photo of

the accident scene; a home insurance claim by submitting a picture of the damage caused by flood, fire and so on; and a health insurance claim by taking a photo of the patient record. The list goes on. If done properly with a high level of accuracy, these applications will not only improve customer satisfaction, they will reduce claim adjustment costs and potentially provide richer information than was available before. Images of accident scenes deliver more information than that provided by adjusters and may also help unify the practices in claim adjustment.

Figure 2
Improving Customer Service with Image Recognition

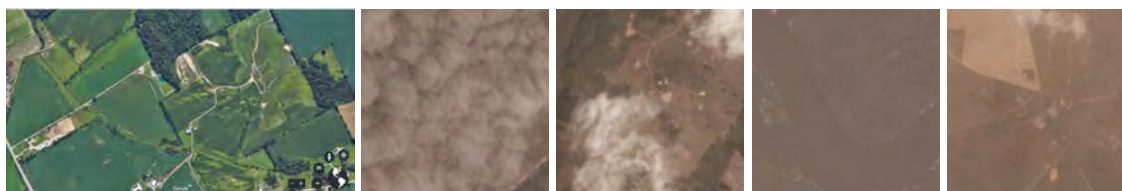


Middle Photo credit: Steve Watts. (<https://pixabay.com/en/car-wreck-accident-crash-insurance-1721724/>).

2. Agricultural insurance pricing and risk analysis can benefit from image recognition techniques. With satellite images such as those shown in Figure 3, these techniques can provide information about the size and type of irrigation, landscape changes and weather conditions for areas as small as several kilometers. This enables more customized and precise prediction of the average crop yield, its volatility and worst case scenarios. Such predictions are unlikely to be achieved without image data, meaning available data for pricing and risk assessment are often at a much higher level.

Figure 3
Satellite Image for Precise Agricultural Insurance Pricing

Pricing factors: size, crop, weather, hydrogeology, landscape and so on.



Map data: Google, DigitalGlobe

3. Property insurance is another area that may benefit from image recognition. Together with the development of the Internet of Things, more images that can provide information on an insured property

will be available. For example, pictures of a house roof can be taken by drones to better understand the condition of the house and determine risk, as seen in Figure 4. Such images can be used not only in the underwriting process but also for loss control, which benefits both clients and the insurance company.

Figure 4

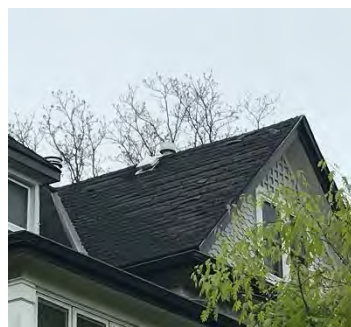
Property Risk Assessment Based on Image Recognition



Good Condition

Risk Level: 2

Risk Loading: -3%



Bad Condition

Risk Level: 6

Risk Loading: 12%

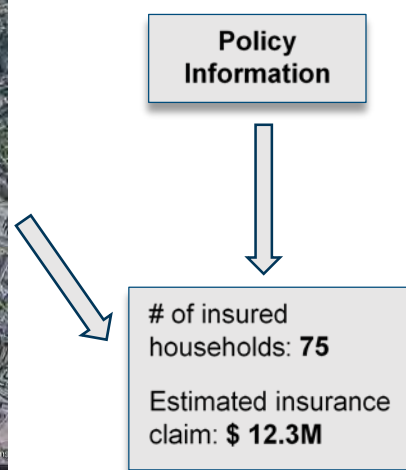
4. For extreme risk events such as tornadoes, hurricanes and wildfires, image recognition techniques can help with real-time risk monitoring and risk management. If a satellite image of a tornado path or flood area is available, image recognition models can help detect the damaged areas as well as the degree of damage. Together with policy information, an automated system can predict the claim count and dollar amount, allowing the company to be better prepared to manage claim risk. The series of gray blocks that run from the middle left to upper right in Figure 5 illustrate the path of a tornado; the black dots indicate insured properties. A well-trained image recognition model can identify the tornado path, determine whether the insured homes are damaged and, if so, how severe that damage is. This kind of estimation can be updated in real time with the latest images if a risk event continues to develop.

Figure 5

Risk Monitoring Using Image Recognition



Map data: Google, DigitalGlobe



- Health insurance is another area where image recognition can provide more insight regarding the risk of individual customers. Physicians can use image recognition for diagnosis and prognosis. In addition to linking doctors' opinions to insurance pricing, given enough image data and claim experience, medical images such as computed tomography (CT) images of cancer patients may be used directly as pricing factors to help underwriting and derive risk loading and insurance premium rate. Figure 6 shows the lung images of two patients. Such images are used to determine whether a patient has lung cancer, how severe it is and how it affects the insurance price. It is definitely not the only pricing factor and needs to be used together with physicians' opinions. However, images may provide richer information than diagnosis results and can help improve the accuracy of the pricing model.

Figure 6

Health Insurance Risk Analysis Based on Image Recognition

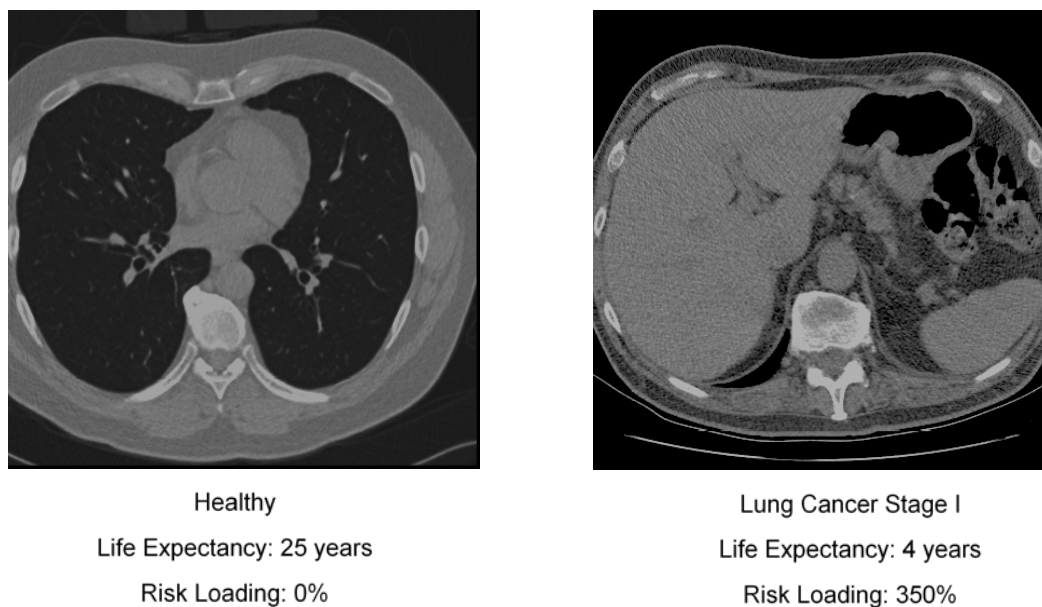


Photo credit: Joanna Shang, 2013.

- Image recognition techniques can also transform hard copy into digital files—often called optical character recognition. This can be very useful for companies that have underwritten business for decades or acquired long-dated business through mergers and acquisitions. Files are scanned, and contents are recognized and can be searched in a computer.

The areas of application mentioned here are far from a complete list. With further improvement of image recognition models and data collection, image data will add more value to the insurance industry. At the current stage, a few factors need to be considered before using automated image recognition techniques for real applications:

- Are there enough data to train the model? Is it easy to collect these data? If the data volume is small, relying on human intelligence may be more efficient and economical.
- Will the use of image data be permitted by customers, regulators and/or the public?
- How much will the image recognition model improve decision making? If the contribution is marginal, there is little financial incentive to use this new technique.

4. How accurate is the automated image recognition? What will be the adverse effects if a prediction is wrong? If a model is not well trained for specific insurance problems, low accuracy and adverse effects could kill the application.

Like the adoption of any new technology, companies need to perform a cost-benefit analysis to examine the investment of resources necessary for and the potential gain and loss from the application. However, this is more a question of *when* image recognition can be applied to a specific business issue rather than *if* the technology should be adopted for the industry as a whole.

Section 3: Challenges

Although the accuracy of image recognition techniques is continually improving, errors are unavoidable. For example, different family members may be recognized as the same person and log in to the same cell phone using facial recognition. This may be a problem if the primary user's insurance account information is protected by facial recognition. Other security measures such as fingerprints may be used with facial recognition to improve security, especially when material policy changes and transactions are involved.

Applying image recognition to the insurance industry still faces many challenges. Existing models are usually trained to identify the objects in an image. However, for insurance applications, more useful information may be the behavior of the object. For example, is a driver using a cell phone while driving? Is a house's roof in good enough condition to withstand hurricanes? How many insured houses were hit by a tornado? This often leads to more customized model training using relevant image data. Data collection and model training may take a long time and require many resources.

Another key obstacle of applying image recognition techniques to the insurance industry is the accuracy level. Currently, the highest accuracy rates for popular image recognition competitions, such as the ImageNet Large Scale Visual Recognition Challenge, are usually between 70% and 90%. This is unlikely to be good enough for many insurance applications. Even with the most advanced image recognition models, false conclusions are not unusual. A common example is that an upside-down vehicle, such as the one shown in Figure 7, may be recognized as a plane.

Figure 7

An Ambiguous Image: Car Versus Plane

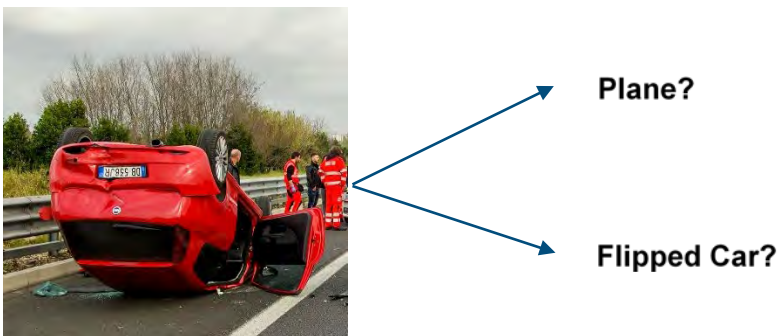


Photo credit: Valter Cirillo (<https://pixabay.com/en/car-accident-clash-rome-highway-2165210/>).

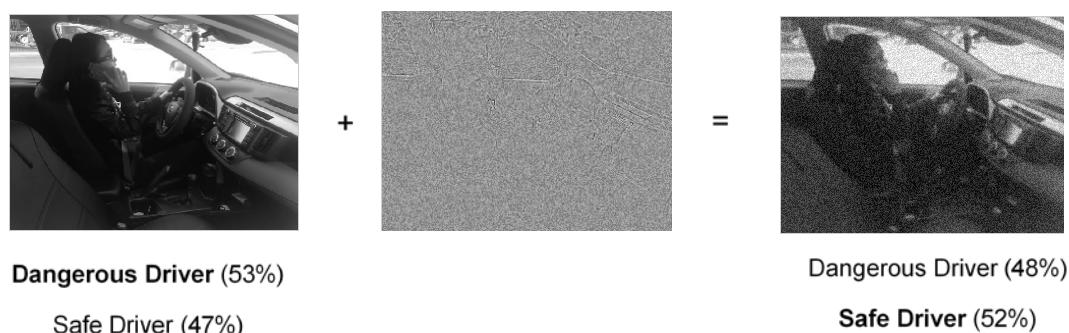
Using an image captioning model that was trained with the Microsoft Common Objects in Context (COCO) image database, the image is interpreted as “a large airplane sitting on the side of a road.” The trained model relies on the shape of the body of the plane (cockpit and fuselage) but omits other parts such as the tail. The model is also unable

to differentiate between a wing and an open door. This can be rectified by feeding more training examples of flipped cars into the model to help it learn the nuances of the shapes. The example also demonstrates the current status of image recognition: a universal model does not exist to meet all the requirements. Fine-tuning the model with specific training data is necessary to improve accuracy for more insurance applications.

The risk of using automated image recognition is not negligible. A small error can lead to customer complaints and financial losses. For example, if a cancer patient applies for life or health insurance and a CT image of the tumor is used to assess the risk and determine appropriate pricing, an error in the model estimation could lead to unnecessarily higher rates or rejection of the application. On the other hand, it could underestimate the risk and specify insufficient rates. If the accuracy rate is not high enough, it could also increase the underwriting risk for the insurance company. At this time, a hybrid approach combining automated image recognition and human intervention is more practical. Human intelligence is still needed to guide and validate automation, especially for high-risk cases and those where the model is less certain.

Cyber risk may also increase when automated image recognition techniques are used. If facial recognition is used to confirm identity, the program may be hacked so that it accepts illegal requests and allows illegal access to private data. Even if the program is safe, carefully designed adversarial examples may be able to fool the recognition model, as demonstrated in Figure 8 with an example from Section 6. By adding some noise to the original image, the prediction changes from a dangerous driver to a safe driver. However, the difference is negligible to human eyes. Models with low accuracy or close probability estimates for different classes are extremely vulnerable to adversarial examples.

Figure 8
Adversarial Example



When using images for fast claim processing, images may be faked or altered to submit false claims. Predictive models must be able to detect these frauds and enhance the defense against adversarial examples.

Regulatory and reputational risk may also occur with the application of image recognition. Image data may not be considered private and thus may be judged inappropriate for determining insurance rate and claim. If an insurance company does not fully explain its methods to customers, regulators and the public, using nonpublic image data could cause reputational incidents. Even if the data are usable, the way they are used to assess risks and determine insurance premium may be questioned. Clients may want to know exactly how the algorithm determines the risk loading for insurance products. Goodman and Flaxman (2016) studied the impact of the European Union's (EU's) general data protection regulation that restricts automated individual decision making that "significantly affect" users. Insurers that want to use image recognition technology need to be prepared for explanation and communication.

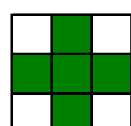
Further developments are needed to improve model accuracy so that models are intelligent enough to improve analysis and decision making. At the same time, such models provide many opportunities for actuaries. Given their strong statistical background, actuaries have already mastered the fundamentals of image recognition models, so they can quickly grasp the new techniques with proper training. Equipped with both the industry knowledge and the technical skills, actuaries can participate in linking image recognition to risk assessment and decision making in a meaningful way. They can also help to design image recognition model structures that can solve more complicated insurance-related issues and validate image recognition conclusions with their existing models based on alternative data sources. This will help to limit the risks of image recognition.

Section 4: Data Processing

When given an image, a model's first task is to transform that image into the formats computers can understand. An image is composed of pixels, the smallest picture element in digital imaging. Each pixel can be described by its color, which can be represented by a color imaging system. The most popular color imaging system is the RGB system. A color can be represented by a three-element vector, including the scales at each color dimension: red, green and blue. For example, white is represented as (0,0,0) and black is represented as (255, 255, 255). A green cross with a white background can be easily transformed to a numeric matrix of three-number vectors, as shown in Figure 9.

Figure 9

Image Data Representation

	=	<table border="1"> <tr> <td>(0,0,0)</td><td>(0,128,0)</td><td>(0,0,0)</td></tr> <tr> <td>(0,128,0)</td><td>(0,128,0)</td><td>(0,128,0)</td></tr> <tr> <td>(0,0,0)</td><td>(0,128,0)</td><td>(0,0,0)</td></tr> </table>	(0,0,0)	(0,128,0)	(0,0,0)	(0,128,0)	(0,128,0)	(0,128,0)	(0,0,0)	(0,128,0)	(0,0,0)
(0,0,0)	(0,128,0)	(0,0,0)									
(0,128,0)	(0,128,0)	(0,128,0)									
(0,0,0)	(0,128,0)	(0,0,0)									

In addition to red, green and blue color channels, other channels such as infrared that cannot be seen by humans also may be available and help to enhance the accuracy of image recognition. With images represented by numbers, image recognition tasks are similar to traditional quantitative analysis, except that the data volume is usually larger and the input data need to be transformed into features that are meaningful for a traditional statistical model.

4.1 Data Transformation

Before an image is fed into a model, it may be altered for different purposes. The data volume of high-resolution images can be daunting. A 640×480 image with RGB channels alone is equivalent to 9,216,000 data points ($640 \times 480 \times 3$). This may be too demanding and impractical even with large computing capacity. Given the purpose of image recognition, the colors may not be very important compared to the relative brightness of each pixel. Images may be converted to grayscale to reduce the data volume by two thirds. In grayscale, a pixel can be represented as a weighted average of the red, green and blue values. Figure 10 shows a transformation from a colorful picture to a grayscale, where $\text{grayscale} = (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$. The weights for each color channel are set to make the transformed picture clearer to the human eye. Other weights may be used for automated image recognition as models rely on the numbers rather than the image for analysis. As shown in Figure 10, the transformation is likely to have immaterial impact on understanding the critical information in the photo: a driver is using a cell phone.

Figure 10

Grayscale Transformation

Original ($640 \times 480 \times 3$)Grayscale ($320 \times 240 \times 1$)

An image may also be compressed to a lower resolution. This can reduce the computing requirements and improve the robustness of the model for low-quality images—an idea that is similar to many practices in the actuarial world. Life insurance premium rates usually vary by age in years instead of age in months or days. Individual policies may be grouped into a few representative model points for certain types of analysis. Various levels of detail are foregone to improve speed and reduce computing and administrative burden. Figure 11 shows the compression of a 640×480 image to a 320×240 image. The difference is not obvious to the human eye at this size, but the data volume is reduced by three quarters.

Figure 11

Image Resolution Downsizing

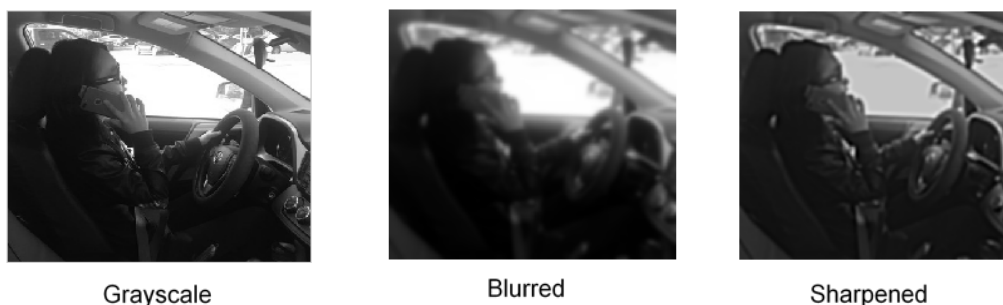
Original (640×480)Compressed (320×240)

In addition to the consideration of data volume, images may be altered to improve the robustness of the model or enhance the accuracy of image recognition. If all the images used for model training are of high quality, the recognition model may not be very useful when only lower quality images are available in certain circumstances. With this thinking, high-quality images may be converted to low-quality images during training to improve the robustness of the resulting model. A common strategy is to adjust each pixel according to the weighted value of its neighbors. The weight of each neighbor depends on its distance from the central pixel using a Gaussian filter. The degree of blurring can be controlled by the volatility parameter in that function. The higher the volatility parameter, the more blurred the altered image. On the other hand, when the accuracy of image recognition cannot be compromised, images may be enhanced to improve model input in the hope of achieving higher accuracy. Images may be sharpened to enhance the edge contrast. The sharpening process is a reverse of the blurring process, applying a function such as a Gaussian filter to the image. Figure 12 shows an image that has been blurred and

sharpened. The grayscale image is blurred with a Gaussian function. The blurred image is then sharpened to create a clearer photo.

Figure 12

Image Blurring and Sharpening

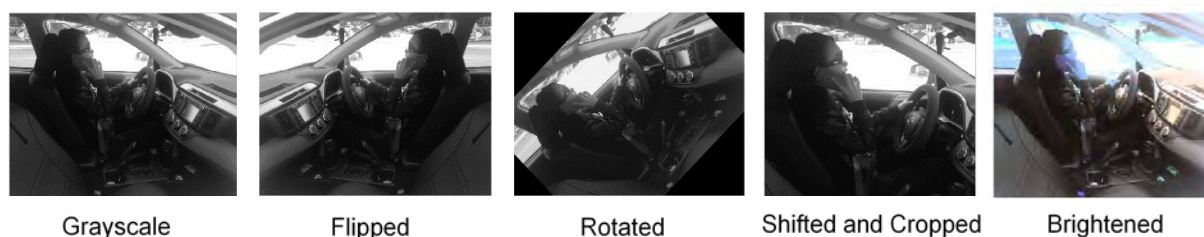


4.2 Data Augmentation

In addition to altering images before feeding them into an image recognition model, one may augment them. Normal augmentations include flipping, rotating, shifting, cropping and brightening. Usually augmentation is either used in additional training examples or to increase the robustness of the image recognition program. If a model is robust, it should be able to generate the same conclusion no matter from what angle the image is taken. Figure 13 shows some augmented images that can be used as additional training data.

Figure 13

Image Augmentation



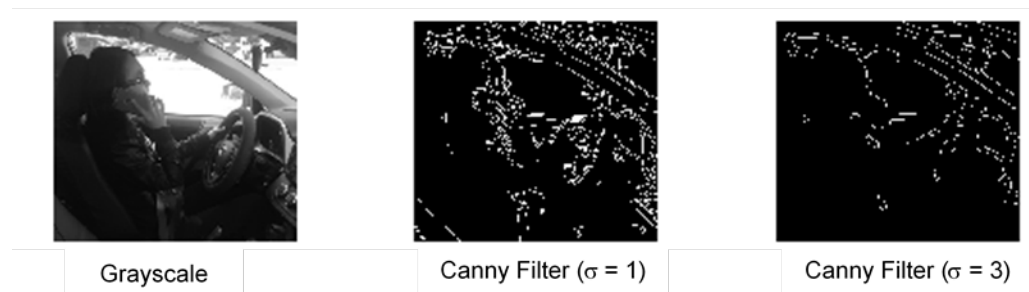
4.3 Feature Extraction

A further step is to detect and extract features from image data. A naïve approach is to use descriptive statistics to represent the image data. For example, the mean, volatility, skewness, kurtosis and percentiles of the pixels in an image can be used as model inputs. This is quite similar to some actuarial analyses in which feature extraction is applied to data that are not readily meaningful for predictive models. For example, GPS tracking data consisting of time and geolocation, average speed, distance, maximum speed and speed percentiles may be calculated and used to assess the risk of driving behavior.

More complex feature detections include measuring the changes from pixel to pixel. Figure 14 shows an edge detection exercise where the areas with the highest contrast are identified by calculating the image gradients at the pixel level and showing the pixels with high gradients. If the goal of the image recognition task is to identify a distracted driver using a cell phone, detected edges may be sufficient to get the conclusion. A rectangle (cell phone) held in one hand with the other hand controlling an oval object (the steering wheel) are the key elements to look for

in this situation. Detected edges, edge orientation and gradient magnitude can be used as additional features for a predictive model. This is analogous to analyzing a driver's fast acceleration and deceleration patterns to find the information that is most useful for risk assessment. Smooth acceleration and deceleration behaviors are less useful.

Figure 14
Edge Detection



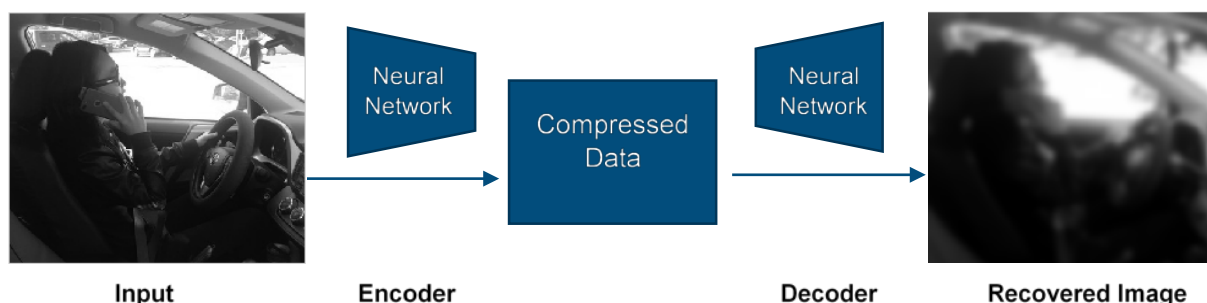
Note: Canny algorithm is used in edge detection, where σ stands for the degree of smoothness before calculating the gradients.

Feature extraction is a manual process requiring a specific algorithm. With the development of deep learning models, features can be learned automatically, as will be discussed in Section 5.2.

4.4 Autoencoders

Autoencoding is another way to reduce data volume. In actuarial analysis, principal component analysis (PCA) or singular vector decomposition (SVD) are used to reduce the amount of data. The idea is to construct a few transformed variables—usually less than five—that contain the majority of the volatility in a dataset. Autoencoding is the equivalent of PCA/SVD in image recognition, with the goal representing images with less data. It transforms image data using an artificial neural network (ANN) called an encoder to get the compressed data. The compressed data are used as model input for image recognition. Compressed data can be used to reconstruct the image using another ANN called a decoder. Figure 15 shows the process of applying an autoencoder.

Figure 15
Autoencoder



How the neural networks can be calibrated? The object is to minimize the difference between the input image and the recovered image. Assuming the input image is a 64×64 grayscale image, a simple autoencoder can be described as follows:

$$\text{Input: } X = [x_1, x_2, x_3, \dots, x_{4096}]'$$

Encoder $C = [c_1, c_2, c_3, \dots, c_f]'$ with f key features

$$C = \frac{1}{1 + e^{-(WX+b)}}$$

$$\text{Decoder } \hat{X} = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \dots, \hat{x}_{4096}]' = \frac{1}{1 + e^{-(W^d C + b^d)}}$$

Where

x_i = the i th pixel of the input image

c_i = the i th feature in the compressed data

W = a $4,096 \times f$ weight matrix

b = a column vector with f elements (It is like the intercept term in a linear function.)

\hat{x}_i = the i th pixel of the reconstructed image

W^d = a $f \times 4,096$ weight matrix

b^d = a column vector with 4,096 elements

Here the ANN has only one hidden layer with sigmoid function as activation function. Each feature in the compressed data C is constructed using a logistic function from the input data. Parameters W, b, W^d , and b^d are calibrated by the difference between X and \hat{X} .

$$\min_{W, b, W^d, \text{ and } b^d} \|X - \hat{X}\|^2$$

All features c_i in the compressed data can be considered the principal components in the PCA. In practice, more complicated ANNs are normally used for encoding and decoding with more hidden layers and other activation functions. The optimization may not be based on minimizing the differences between the input image and the output image but on other objectives such as the distribution of the encoder C . However, the goal remains the same: preserving the key information with less data.

Section 5: Model

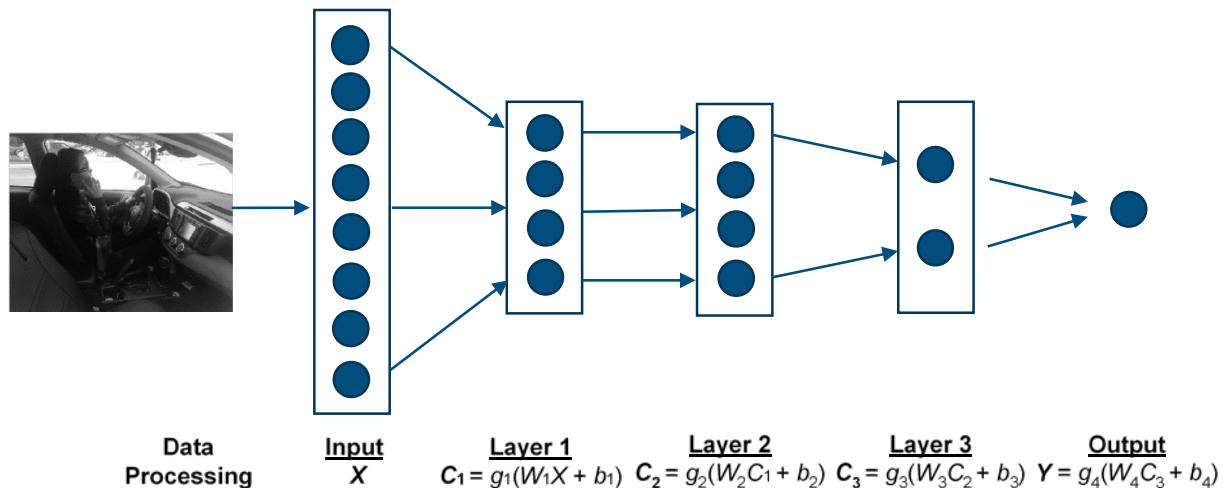
The next step is to feed processed data into the model to retrieve useful information. Many successful image recognition models such as DenseNet, VGGNet, GoogLeNet and ResNet—mentioned briefly in Section 6—are developed from convolutional neural networks (CNNs). Image recognition models have many concepts similar to those of traditional actuarial models. However, image recognition models are more complex, using many traditional statistical models as building blocks. Image recognition also uses the connection of input data, such as their relative positions in an image, in a unique way. The use of parallel computing makes calculation and calibration different. This section explains the major differences.

5.1 Fully Connected Neural Network

Fully connected (FC) neural networks (NNs) were used for image recognition before CNNs became popular. A simple form of FC NN, which is used in autoencoders for data processing, was briefly introduced in Section 4.4. Figure 16 shows a more complicated NN model for image recognition. The model input is 64×64 grayscale images with each represented as 4,096 data points. Three hidden layers are included: layer 1 ($4,096 \times 256$ with 4,096 input data and 256 neurons); layer 2 (256×256); and layer 3 (256×64). The output is the probability that the driver in an image is using a cell phone.

Figure 16

Fully Connected Neural Network



In this case,

C_i = the value of neurons in hidden layer i

g_i = the activation function for hidden layer i

g_4 = the function for the output layer

A common activation function is the sigmoid function $\frac{1}{1 + e^{-x}}$ (aka logistic function). Choices of activation functions are discussed in Section 5.3.

The NN can still be represented as a function of the input X . This is not a simple linear, polynomial, generalized linear or other nonlinear function, but a few layers of linear ($WX + b$) and nonlinear (activation) function stacked together.

$$Y = f(X; W_1, b_1, W_2, b_2, W_3, b_3, W_4, b_4)$$

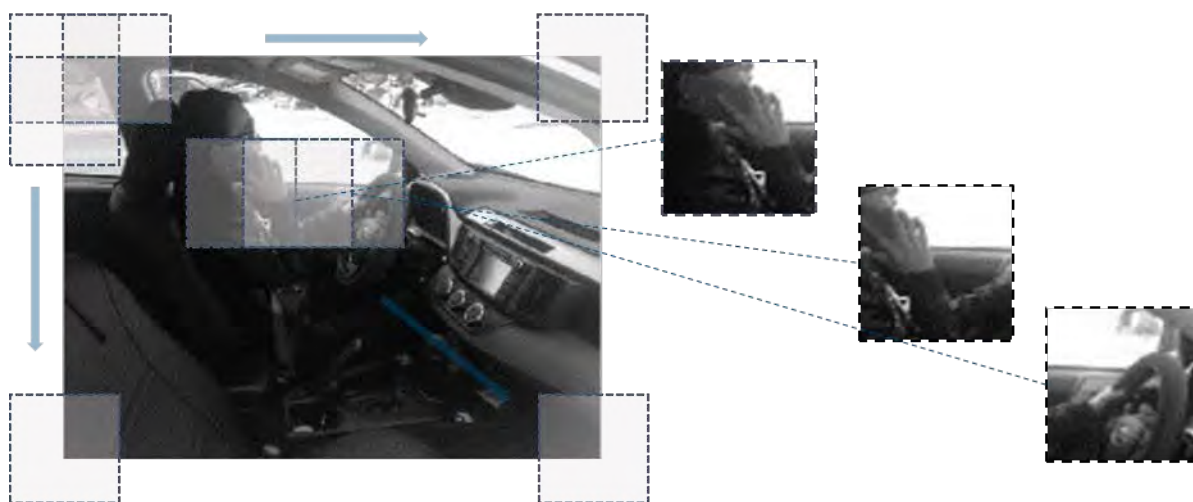
Why are neural networks, rather than traditional statistical models, so important in image recognition? In actuarial analysis, the form of the relationship is normally predefined, such as a linear or nonlinear equation. However, these predefined relationships are not sophisticated enough for image recognition tasks. It is hard to imagine a single pixel's role or weight in determining an image's content. Their combinations, relative locations and contrasts are more important features that cannot be captured easily by traditional statistical models. In addition, with the numerous connections among neurons, NNs can replicate any relationship approximately. A neural network model can be thought of as a high-level model with traditional statistical models as its components.

However, a complicated model has its price as well. The number of parameters is daunting. The model in Figure 16 has more than 1.1 million parameters. It could take a long time for the model to converge, and it requires significant computing resources. It may also cause overfitting when the training data is not sufficient. Another disadvantage of FC NNs is that the relative positions of the data points (pixels) in an image are not used for predicting. The omission of this valuable information significantly reduces the accuracy of the model.

5.2 Convolutional Neural Network

Convolutional neural networks address one important issue in FC NNs for image recognition by reflecting the local connectivity of pixels. How can this be achieved? First, instead of looking at the entire image at the same time, as a fully connected neural network does, a CNN looks at one small area (receptive field) at a time. This is similar to the human way of searching for a small object in a big picture and is illustrated in Figure 17. A small, fixed-size filter runs through the entire image to identify the object or information of interest. It moves with a fixed stride each time, either horizontally or vertically. Three receptive fields comprising a cell phone, the driver's hands and the steering wheel are most useful. The idea of using small receptive fields is common in actuarial analysis. For example, when setting the rate for auto insurance policies, location is an important pricing factor. Even though national data may be available, the most relevant information is from local experience data based on zip code, city or even more granular geolocation.

Figure 17
CNN Receptive Fields



Another advantage of using receptive fields instead of the entire image is the reduction in model parameters. The same parameters are shared among mini-patches, which can reduce training time and the chance of overfitting.

Even using smaller areas, how can the model decide whether they contain the objects of interest? If the goal is to know whether the driver is using a cell phone, the model must determine whether the image contains a hand holding a cell phone with a steering wheel nearby. The model would want to identify rectangles (the cell phone), some connected lines or curves (the hand), and an oval (the steering wheel). It will not look for features such as a blank area and a cross. This is similar to the variable selection in a traditional statistical analysis. Many explanatory variables are available and need to be assessed for their usefulness in predicting the outcome.

However, a key question is how these meaningful explanatory variables can be constructed from image data. In actuarial analysis, this is usually a human-driven process. For example, when analyzing driver behavior from the GPS data of a car, meaningful variables such as speed, acceleration and driving duration can be calculated based on the

past geolocations of the vehicle. Speed can be calculated as distance divided by the time interval, and distance can be calculated based on the starting and ending geolocations. This feature construction process is driven by our understanding of how driving behaviors are measured. In an image recognition task, the important features are not obvious and are difficult to represent with numbers. A desirable explanatory variable in the image recognition example is the likelihood that a cell phone is in the image. How can the image data be altered numerically to express the likelihood? The answer is not obvious to the human mind but can be learned by the CNN.

Figure 18
CNN Structure

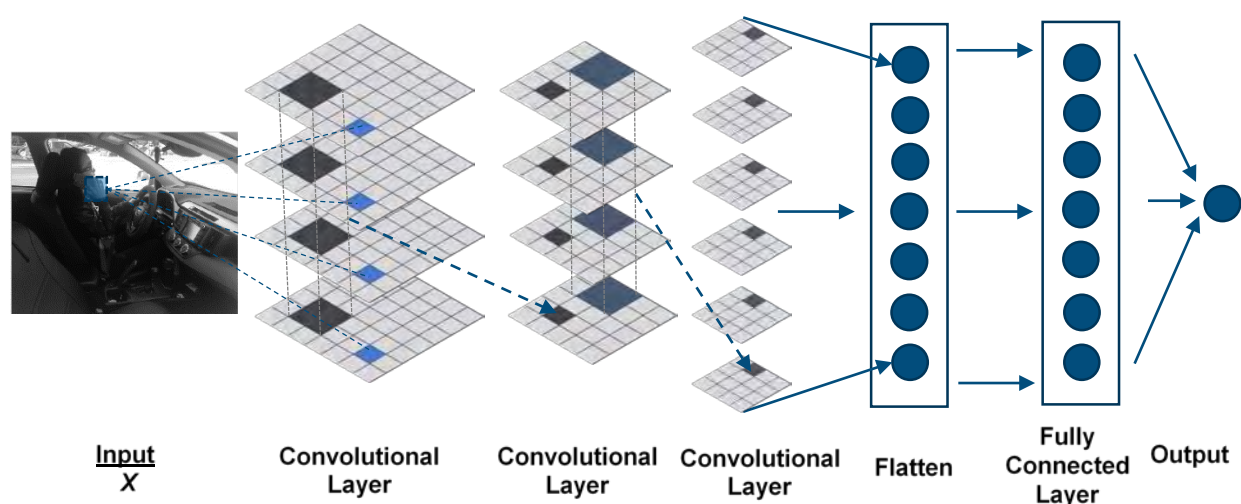


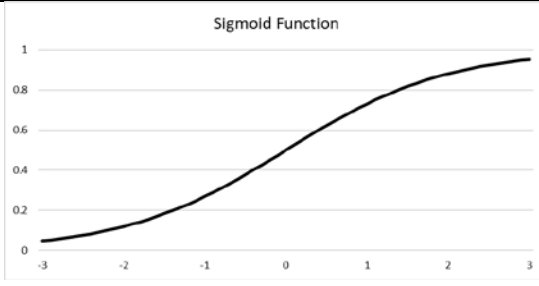
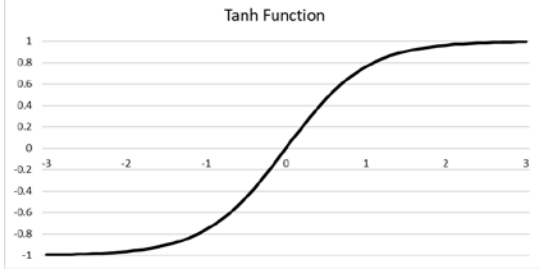
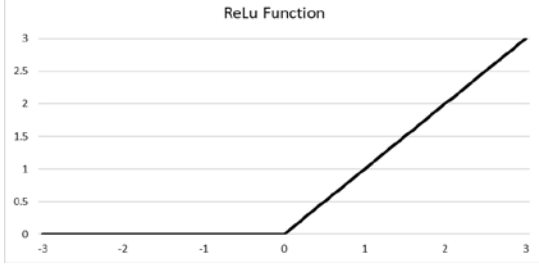
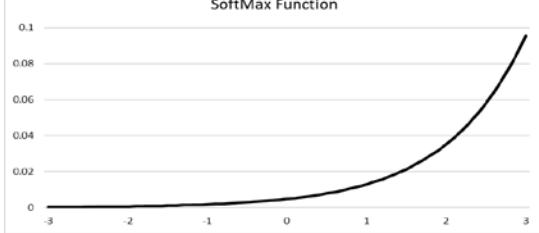
Figure 18 shows the structure of a simple CNN. Input data are mapped into feature maps by the discrete convolution of image patches and kernels of the same size. Unlike a fully connected neural network in which neurons in the same layer are separate, each CNN feature map is a two-dimensional matrix that keeps local connections when forming the features. A kernel is similar to the concept of a weight vector used in an FC NN. However, a kernel is represented as a matrix and applied to patches instead of to the entire image. A CNN usually contains several convolutional layers learning from basic details to high-level abstraction. After that, the feature maps in the final convolutional layer are flattened to a vector of neurons as a hidden layer in an FC NN. Output is then predicted from the fully connected layer.

Convolutional layers are the key to translating image data into useful explanatory variables. The process of predicting the explained variable is no different from that of a traditional statistical model such as logistic regression. A CNN combines data preparation and predictive modeling, and they are driven entirely by training data. Sections 5.3 through 5.6 explain specific components of CNN models in detail.

5.3 Activation Function

In either an FC NN or a CNN, an activation function is applied to a neuron's value before it is fed into the next layer. A neuron's value is a linear function of the neurons in the previous layer ($WX + b$). Depending on its value, the next step is to determine whether the neuron should be activated or not, similar to the way the human brain works. However, the value of the neuron could range from negative infinity to positive infinity and make the choice hard. Activation functions can be considered mechanisms to bring the range down to a manageable level. Many activation functions are available for NNs; Table 1 lists four of the most common ones.

Table 1
Activation Functions

Name	Function	Output Range	Plot
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	(0,1)	 <p>The plot shows the Sigmoid Function, which is an S-shaped curve. The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 1. The curve starts near 0 for negative x, passes through (0, 0.5), and approaches 1 for positive x.</p>
Tanh	$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$	(-1,1)	 <p>The plot shows the Tanh Function, which is an S-shaped curve centered at the origin. The x-axis ranges from -3 to 3, and the y-axis ranges from -1 to 1. The curve passes through (0, 0) and approaches -1 for negative x and 1 for positive x.</p>
ReLU	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$[0, \infty)$	 <p>The plot shows the Rectified Linear Unit (ReLU) Function. The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 3. The function is zero for negative x and increases linearly for positive x.</p>
SoftMax	$f(x_j) = \frac{e^{x_j}}{\sum_{i=1}^N e^{x_i}}$ for $j = 1, 2, \dots, N$ $\sum_{i=1}^N f(x_i) = 1$	[0,1]	 <p>The plot shows the SoftMax Function. The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 0.1. The function is zero for negative x and increases exponentially for positive x, approaching 1 as x increases.</p>

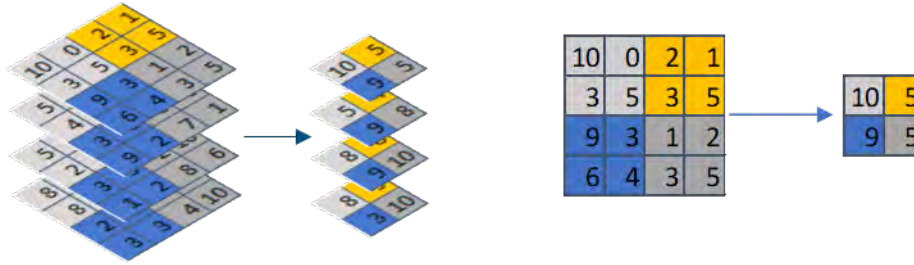
The choice of activation function can be arbitrary and is often by trial and error. To facilitate the training process that involves gradient calculation, the activation function needs to be smooth. When the output is a probability, sigmoid function is a natural choice, as in logistic regression. The ReLU function is less smooth than some others but works well in practice.

5.4 Pooling

In addition to convolutional and fully connected layers, pooling layers are a common layer type used for image recognition in CNNs. Pooling is basically a down sampling method that merges a few pixels into one. Figure 19 shows a popular pooling method called max pooling. It uses the maximum of a subset to replace the subset in the next layer. For example, a 4×4 matrix becomes a 2×2 matrix. The benefit of using a pooling layer is to reduce the calculation and data dimension.

Figure 19

Max Pooling



Other pooling methods, such as average pooling using the average value and stochastic pooling using the value of a randomly chosen element, can also be used.

5.5 Normalization

In some actuarial analysis, input data is normalized to the same range before a statistical model is applied. Data may be scaled into the range of (0,1) using its minimum and maximum, such as $\frac{X - X_{min}}{X_{max} - X_{min}}$. The data may also be standardized using the mean and volatility, as in $\frac{X - \mu_X}{\sigma_X}$. Data normalization is important for two reasons:

1. Some data fields may have a much larger magnitude than others. This will give the larger fields unfair weight, especially in classification tasks where distances among data points are calculated and used to determine similarity. By normalizing the data, the dominating impact of large fields can be reduced.
2. Model training will be faster with normalized data using a gradient descent algorithm.

As in traditional statistical analysis, data normalization is important for image recognition due to the difficulty of model optimization. Unlike the traditional data normalization methods that put data in a desired range, NNs can allow the model to learn about the appropriate range for itself. Ioffe and Szegedy (2015) suggested the following data normalization method based on mini-batch data:

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

Where

x_i = the i th input element in the mini-batch X

y_i = the i th output element from the mini-batch X after the normalization batch layer

μ_B = the mean of input X in a mini-batch

σ_B = the standard deviation of input X in a mini-batch

ϵ = a small constant to maintain stability

γ, β = parameters to be calibrated and used for all mini-batches

Parameters γ and β control the target range after normalization for the entire dataset. CNN models converge faster with normalization layers.

5.6 Regularization

In statistical analysis, when the data volume is limited, the calibrated model may overfit the data. The number of model parameters is too large relative to the input data. To avoid this problem, traditional statistical models such as linear regression are modified to penalize the additional model complication. Ridge regression and the least absolute shrinkage and selection operator (LASSO) are two examples of regularization used in actuarial analysis. Ridge regression uses the sum of squared model parameters as the penalty (L2 regularization), whereas the LASSO uses the sum of the model parameters' absolute value (L1 regularization) as the penalty:

$$\text{Ridge Regression: } \min_{\beta} \{(Y - \beta X)^T (Y - \beta X) + \lambda \|\beta\|^2\}$$

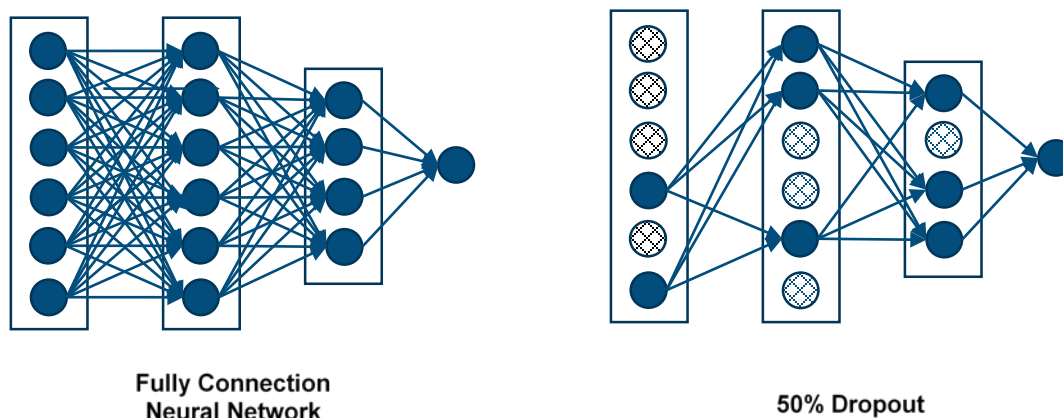
$$\text{LASSO: } \min_{\beta} \{(Y - \beta X)^T (Y - \beta X) + \lambda \|\beta\|\}$$

Neural networks are vulnerable to overfitting problems given the large number of parameters and sometimes limited training data. Both L1 and L2 regularization can be applied to neural networks by adding a penalty component in the objective function according to the value of the parameters (W , b , γ , β and so on), as in ridge regression and the LASSO.

Another regularization approach for image recognition is adding a dropout layer to the neural network. Srivastava et al. (2014) introduced the method of dropping a percentage of neurons into the network during model training. The dropped-out neurons are randomly chosen and their value set to zero. The remaining neurons are scaled up by $1/(1 - p)$, where p is the percentage of neurons dropped out. The idea is similar to the Random Forest model in which each tree can choose from only a random subset of features to construct the decision rules. Figure 20 shows a fully connected NN compared to a 50% dropout NN.

Figure 20

Dropout Regularization



Traditional regularization methods and dropout can be used together for image recognition models.

5.7 Calibration

Connection neural networks can be structured differently by stacking all the layers together in various sizes, orders and times. The next step is to calibrate the model. The goal is similar to that of most predictive models: minimizing the difference between predicted results and the reality with a regularization term. The following objective function is the same as the one for ridge regression except that the function f is more complicated than a linear function.

$$\min_{W,b,\gamma,\delta} L = \min_{W,b,\gamma,\delta} \|Y - f(X; W, b, \gamma, \delta)\|^2 + \lambda \| [W, b, \gamma, \delta] \|^2$$

As in many statistical models, the gradient descent method is used to minimize the loss function. A model parameter p is updated gradually in the optimization process until the loss function stops decreasing:

$$p = p - \alpha \frac{\partial L}{\partial p} \quad p \in [W, b, \gamma, \delta]$$

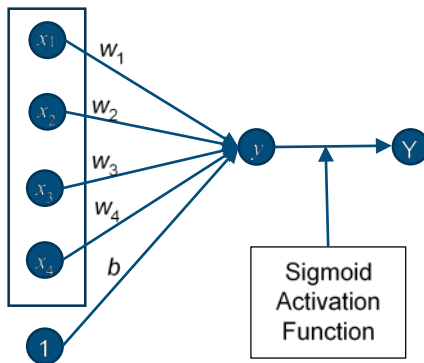
Where

α = the learning rate that controls the updating speed

Compared to closed-form solution to linear regression, the complication of using gradient descent for neural network optimization is the calculation of the gradient $\frac{\partial L}{\partial p}$ because the layers are stacked together. The backpropagation algorithm can be used to calculate the gradient using the chain rule in reverse. Figure 21 shows the backpropagation for a logistic model represented as a fully connected layer with a sigmoid activation function.

Figure 21

Logistic Regression as a Neural Network



Fully connected layer output: $y = b + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$

Sigmoid activation function: $Y = \frac{1}{1 + e^{-y}}$

This is equivalent to the logistic model $Y = \frac{1}{1 + e^{-(b + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4)}}$

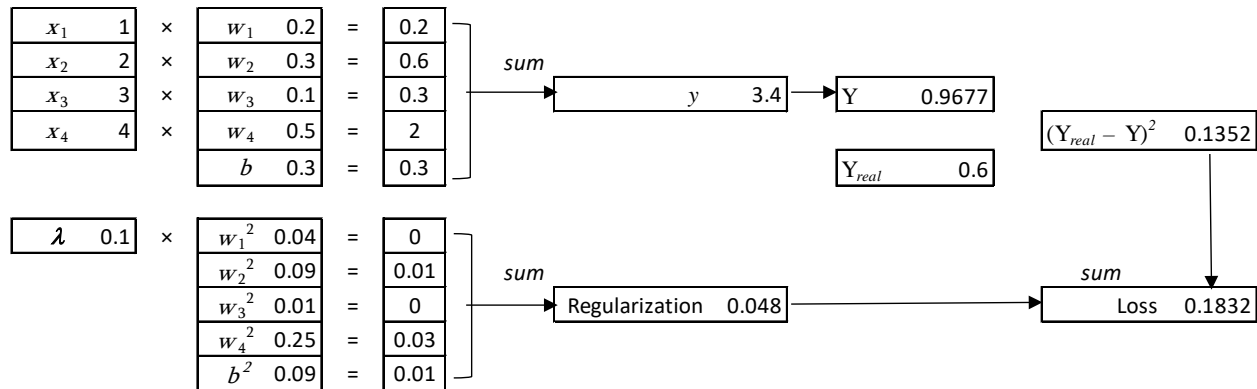
Loss function:

$$L = \frac{1}{m} \sum_{i=1}^m [Y^i - f(X^i; W, b)]^2 + \lambda (b^2 + \sum_{j=1}^4 w_j^2)$$

For illustrative purposes, we assume that only one training data record is available ($m = 1$) and the model parameters are randomly generated. Figure 22 shows the forward pass from the input to the loss function.

Figure 22

Initial Forward Pass



With all the values in the forward pass, the gradient $\frac{\partial L}{\partial p}$ can be calculated backward. For example, the calculation of $\frac{\partial L}{\partial w_1}$ can be done in the following steps:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial y} \frac{\partial y}{\partial w_1} + 2\lambda w_1$$

$$\frac{\partial L}{\partial Y} = \frac{\partial (Y_{real} - Y)^2}{\partial Y} = -2(Y_{real} - Y) = -2(0.6 - 0.9677) = 0.7354$$

$$\frac{\partial Y}{\partial y} = \frac{\partial \frac{1}{1 + e^{-y}}}{\partial y} = \frac{e^{-y}}{(1 + e^{-y})^2} = Y(1 - Y) = 0.9677(1 - 0.9677) = 0.0313$$

$$\frac{\partial y}{\partial w_1} = \frac{\partial (b + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4)}{\partial w_1} = x_1 = 1$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial y} \frac{\partial y}{\partial w_1} + 2\lambda w_1 = 0.7354 \times 0.0313 \times 1 + 2 \times 0.1 \times 0.2 = 0.0630$$

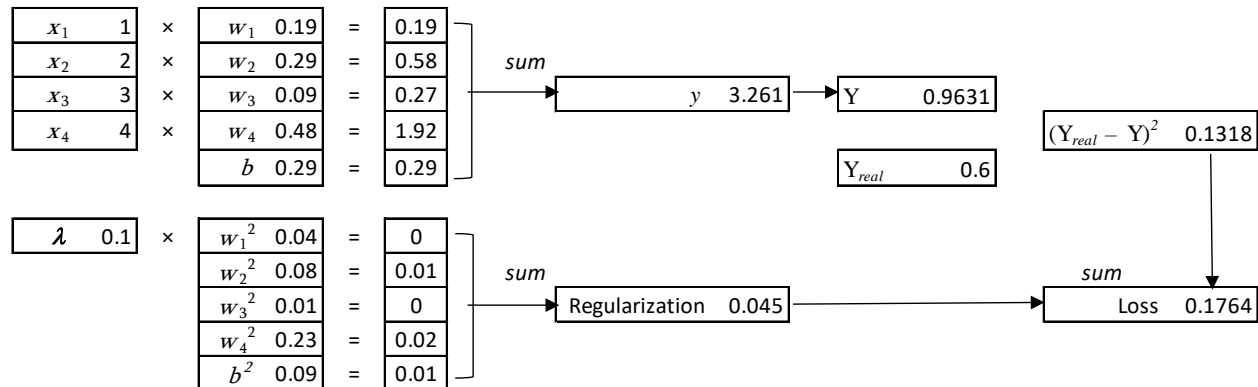
For the next iteration, w_1 can be updated as

$$w_1 = w_1 - \alpha \frac{\partial L}{\partial w_1} = 0.2 - 0.1 \times 0.0630 = 0.1937$$

After updating all the parameters, the loss declines from 0.1832 to 0.1764, as shown in Figure 23.

Figure 23

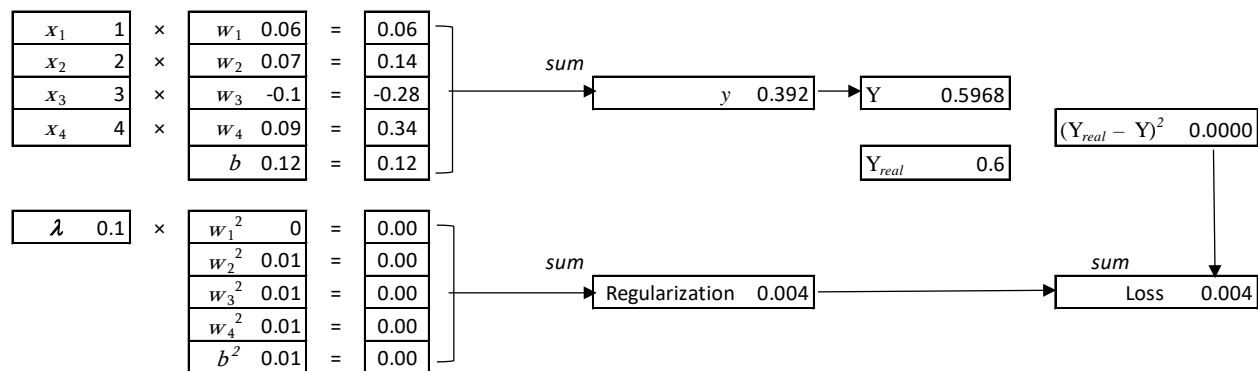
Forward Pass after the First Round of Updating



After 28 rounds of updating with a learning rate of 0.1, the gradient descent method with backpropagation finds the optimal parameter values with a loss of 0.0040, as shown in Figure 24.

Figure 24

Forward Pass after the 28th Round of Updating



This example can easily be expanded with more training data when the average of the prediction errors is used in the loss function. For neural networks with more layers, the backpropagation method uses the chain rule from the loss function backward through layers to calculate the gradients and update all the parameters.

Even though backpropagation is an efficient way of updating parameters in the optimization process, a large volume of training data still presents a challenging task, considering the number of parameters that a CNN can have. A solution is to update the parameters based on a subset of training data instead of the entire dataset. For example, if the dataset has 25,600 training examples, the parameter updating process needs 100 iterations to run through the entire dataset, with each iteration using only 256 training examples. In each iteration the loss function is calculated based on only 256 training examples. This can materially reduce the training time. An extreme case of batch-based

gradient descent is stochastic gradient descent (SGD), where the parameters are updated with one training example at a time. Technology development such as parallel computing, graphics processing unit (GPU) also enables the calibration of image recognition models.

Based on the preceding discussions, image recognition models may be presented and aggregated in a different way from traditional statistical models. However, the concepts and building blocks of image recognition are not greatly dissimilar to actuarial models.

Section 6: Example: Driver Behavior Assessment Using Image Recognition

This section presents an example of using image recognition application to assess driver behavior. Given images of people behind the wheel, we want to know whether drivers are concentrating on driving or not. The process of data processing, model setup, prediction and model validation are described. Codes used in this example have been made public for educational purposes and are hosted at GitHub with documentation (<https://github.com/windwill/imgRecognition>).

6.1 Data Processing

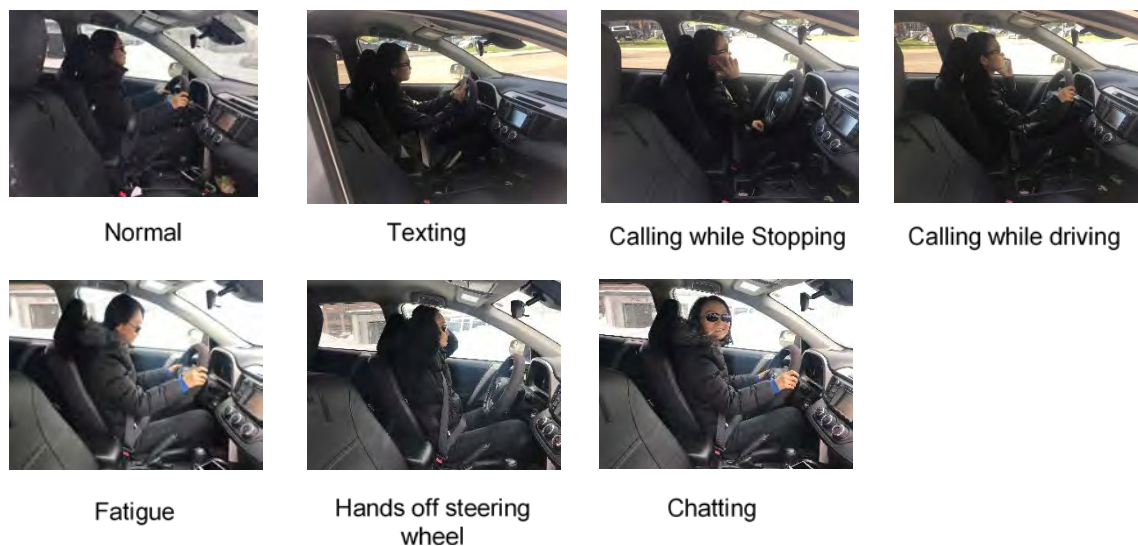
The dataset includes 12,000 images of drivers labeled as one of the following:

- Normal driving
- Texting
- Talking using cell phone
- Hands off steering wheel
- Fatigue
- Seatbelt off
- Chatting with eyes off the road

Figure 25 shows examples of the images in the dataset.

Figure 25

Data Example



The images are JPG files with three color channels: red, green and blue. Since the dataset volume is not large, the images have not been transformed to grayscale. The images in the dataset are well centered and in good direction. Therefore, only one image augmentation is used to enlarge the training data volume: left-right flip.

6.2 Model

In this example, a few variations of deep and complicated CNN models are used, including VGG, GoogLeNet, ResNet and DenseNet. They are introduced here according to ascending numbers of layers.

VGG16

VGG models, also known as very deep convolutional networks proposed by Simonyan and Zisserman (2015), use more layers with small filters to construct deeper models. VGG16 is a model that uses 16 layers.

GoogLeNet (Inception3)

GoogLeNet is a 22-layer CNN model proposed by Szegedy et al (2014). Unlike standard CNN models that have only one layer following the previous layer, GoogLeNet uses the Inception module to allow parallel operations. The model is much deeper than VGG16 but also has many more parameters to fit.

ResNet50

He et al (2015) proposed ResNet using residual blocks to improve optimization. Instead of computing the features as layer output, ResNet adds the layer input to the layer output as the final output to keep complete information after layer transformation. ResNet 50 is a 50-layer model.

DenseNet121

Huang et al (2018) proposed DenseNet, also known as a densely connected convolutional network. DenseNet allows not only sequential connection among layers, but also direct connection between layers that are not neighbors. DenseNet121 has 121 connected layers.

The number of layers and the number of neurons in each layer depend on the balance between accuracy and training speed. More layers and neurons may lead to higher accuracy but low training speed and require more computing time and resources if the model is applied in the real world. Available computing capacity may also place a limit on model complexity. When training examples are not enough, models that are too complicated may lead to overfitting and false high accuracy. It is usually a trial and error process to determine the most appropriate model.

6.3 Training

Many frameworks such as Caffe and Caffe2, Deeplearning4j, Microsoft CNTK, MXNet, Keras, Torch, PytorchGoogle TensorFlow and Theano can be used to train image recognition models without significant programming efforts. (A detailed comparison of popular deep learning tools can be found at

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software.) According to a study performed by Intel AI Academy (2017), TensorFlow is the most popular deep learning framework followed by Caffe, Keras, Microsoft CNTK, MXNet and Torch. Popularity is assessed based on GitHub repository stars that are used by programmers to keep track of interesting projects and Stack Overflow questions that asked by programmers. The choice of deep learning framework depends on many other factors such as programming language, ease of scaling, computing resource requirements, availability of pretrained models, debugging and support of parallel computing. In this example, PyTorch is used for training the models. It is an open-source Python version of the Torch framework that

contains many deep learning models and has the infrastructure to use GPU for computation. Pytorch is used here purely because of the author's personal preference.

Table 2 lists some training parameters used in this example. Explanations follow the table.

Table 2
Model Training Parameters

Parameter	Value
Batch Size	72
Image size (pixel)	256 × 256
Epoch	30
Learning rate	0.1 for the first 10 epochs, 0.01 for the next 15 epochs and 0.005 thereafter
Optimization method	Stochastic gradient descent
Loss function	Cross-entropy loss
Accuracy	F-Measure
Pretrained	ImageNet pretrained models

Batch Size

The model is trained using 80% of the entire dataset, with the rest used for model validation. Given the large volume of data an image carries, it is impossible to use all the training images together for optimization. Rather, the optimization is done many times with smaller datasets referred to as batches. For example, if each batch contains 72 images, running through the entire training dataset (9,600 images) means running 133 batches. For each batch, the model tries to improve the accuracy and reduce the loss.

Image Size

Images are usually resized to a resolution based on the computing capability, prediction target, and trade-off between speed and accuracy. In this example, images are resized to 256 × 256.

Epoch

One epoch means the entire training dataset will be run through one time. Usually the training dataset needs to be used multiple times for the optimization process to converge. In this example, 30 epochs are used.

Learning Rate

Learning rate controls the parameter updating speed during training. A high learning rate means the gradient information will be reflected at a higher speed.

Optimization Method

In this example, stochastic gradient descent is used to minimize the loss function. It approximates the gradient descent in a stochastic way by using single training examples to update model parameters. Other optimization methods can be used as well.

Loss Function

In most actuarial models, the loss function or error function is based on the difference between the estimated value and the actual value. For example, the mean squared error (MSE) is the average of squared errors. MSE works perfectly for regression but not for classification. In this example, the actual value could be either 1 or 0, whereas the estimated value is usually a probability. Multiclass classification makes the loss measurement even more complicated. Cross-entropy loss is a common loss measure for multiclass classification.

$$-\sum_{c=1}^L I_{actual,c} \log(p_{actual,c})$$

Where

c = the label of an image, such as normal, chatting, texting and so on

$I_{actual,c}$ = an indicator function that equals one when an image's actual label = c and zero otherwise

$p_{actual,c}$ = predicted probability that the image has a label c

Cross-entropy loss increases when the predicted probability decreases.

Accuracy

In this example, F-measure is used to measure the model accuracy. Precision measures the type I error, and recall measures the type II error. F-measure is the harmonic average of precision and recall.

Table 3

Confusion Matrix

	Predicted: Positive	Predicted: Negative
Actual: Positive	True positive	False negative
Actual: Negative	False positive	True negative

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

F-measure is calculated for each label separately and then averaged to get the aggregate value.

Pretrained Model

For business applications of image recognition techniques, the most common obstacle is insufficient data volume. By training from scratch, models may converge too slowly or unsuccessfully. The risk of overfitting may also exist when data volume limits the credibility of the validation. This can be true for insurance applications as well, especially in the early stages of application.

To overcome this issue, general pretrained models can be used as the starting point for further training. These models use datasets such as ImageNet that contains millions of images. While these general object images may not be directly related to insurance problems, using pretrained models can improve convergence speed and mitigate overfitting. Using pretrained models, it is like asking a child who is able to recognize general objects to learn more specific skills such as understanding what the objects are doing. It is much easier than asking an infant to learn the behaviors of objects even if the infant has not learnt about recognizing the objects.

6.4 Validation

A validation dataset is used to test the accuracy of each trained model. Table 4 lists the cross-entropy loss and aggregated F-measure for each model.

Table 4
Validation Result

Model	Cross-Entropy Loss	F-Measure
VGG16	1.19	0.571
DenseNet121	0.096	0.970
GoogLeNet (Inception3)	0.162	0.956
ResNet50	0.153	0.952
Ensemble*	0.151	0.976

*Ensemble uses average probability estimates of DenseNet121, Inception3 and ResNet50. VGG16 is not included because of its lower accuracy compared to the other models.

The prediction accuracy is high except for VGG16, which may indicate a certain level of overfitting. The dataset contains similar pictures regarding the shooting angle, human gestures, brightness and background. The accuracy level may drop for images taken in a different environment. This is a common challenge for image recognition applications when the volume of data is insufficient. It cannot be solved by models—only with more data.

6.5 Application

Although the purpose of demonstrating this example is for education, models may be used for refining the pricing of commercial auto insurance and personal auto insurance. Image recognition results such as the probability of using phones while driving can be used as additional pricing factors. However, data availability could be a problem. For drivers with good attentions on driving, it could mean an insurance rate discount. For drivers with unsafe behaviors, it could mean an insurance rate hike. Therefore, only clients with good driving behaviors would want to participate in such a program. This makes the benefit of image recognition diminishing for this insurance application.

A more practical way to encourage clients' participation is to present it only as a discount program. Discounts will be given for safe driving while unsafe behaviors will not cause denial of insurance application or rate increase above

previous benchmark. When unsafe behaviors are detected using the trained model, real-time warning such as a beep or a voice notification may be triggered to encourage safer driving.

This shows the potential role of image recognition in insurance pricing, risk assessment and risk mitigation at a more personalized level.

Section 7: Conclusion

Image data has become another source of information that can potentially benefit the insurance industry. Image recognition models such as CNN are built on traditional statistical models but are more complicated because they stack simple models together. However, the methods of image data processing and image recognition are not too different from those of traditional models.

In the insurance industry, image recognition techniques can help improve customer service, facilitate data collection and provide additional valuable information for pricing, reserving and risk management. On the other hand, decision making supported by automatic image recognition faces certain challenges. Models need to be fine-tuned with insurance-related image data. Meaningful model output for insurance applications need to be more sophisticated than object identification. Model accuracy can still be an obstacle for certain applications. Fraud detection is also necessary to avoid the use of fake or adversarial examples. At the same time, these challenges provide opportunities for actuaries to improve the business value of using image recognition techniques to solve insurance problems.

References

- Goodman, Bryce, and Seth Flaxman. 2016. "European Union regulations on algorithmic decision-making and a 'right to explanation.'" <https://arxiv.org/pdf/1606.08813.pdf>.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 2015. "Deep residual learning for image recognition." <https://arxiv.org/pdf/1512.03385v1.pdf>.
- Hoerl, A.E., and R. Kennard. 1970. "Ridge regression: Biased estimation for nonorthogonal problems." *Technometrics* 12(1): 55–67.
- Huang, Gao, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger. 2018. "Densely connected convolutional networks." <https://arxiv.org/pdf/1608.06993.pdf>.
- Intel AI Academy. 2017. "Hands-on AI part 5: Select a deep learning framework." <https://software.intel.com/en-us/articles/hands-on-ai-part-5-select-a-deep-learning-framework>.
- Ioffe, Sergey, and Christian Szegedy. 2015. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." <https://arxiv.org/pdf/1502.03167.pdf>.
- Kappagantula, Srilatha, and Amol Kulkarni, 2017. "Future of image technologies in financial services" [white paper]. <https://www.infosys.com/industries/financial-services/white-papers/Documents/future-image-technologies.pdf>
- Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. 2014. "Dropout: A simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research* 15: 1929–1958.
- Tibshirani, Robert. 1996. "Regression shrinkage and selection via the Lasso." *Journal of the Royal Statistical Society* 58(1): 267–288.
- Simonyan, Karen, and Andrew Zisserman. 2015. "Very deep convolutional networks for large-scale image recognition." Presented at the International Conference for Learning Representations, <https://arxiv.org/pdf/1409.1556.pdf>.
- Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. "Going deeper with convolutions." <https://arxiv.org/pdf/1409.4842.pdf>.

About The Society of Actuaries

The Society of Actuaries (SOA), formed in 1949, is one of the largest actuarial professional organizations in the world dedicated to serving 30,000 actuarial members and the public in the United States, Canada and worldwide. In line with the SOA Vision Statement, actuaries act as business leaders who develop and use mathematical models to measure and manage risk in support of financial security for individuals, organizations and the public.

The SOA supports actuaries and advances knowledge through research and education. As part of its work, the SOA seeks to inform public policy development and public understanding through research. The SOA aspires to be a trusted source of objective, data-driven research and analysis with an actuarial perspective for its members, industry, policymakers and the public. This distinct perspective comes from the SOA as an association of actuaries, who have a rigorous formal education and direct experience as practitioners as they perform applied research. The SOA also welcomes the opportunity to partner with other organizations in our work where appropriate.

The SOA has a history of working with public policymakers and regulators in developing historical experience studies and projection techniques as well as individual reports on health care, retirement, and other topics. The SOA's research is intended to aid the work of policymakers and regulators and follow certain core principles:

Objectivity: The SOA's research informs and provides analysis that can be relied upon by other individuals or organizations involved in public policy discussions. The SOA does not take advocacy positions or lobby specific policy proposals.

Quality: The SOA aspires to the highest ethical and quality standards in all of its research and analysis. Our research process is overseen by experienced actuaries and non-actuaries from a range of industry sectors and organizations. A rigorous peer-review process ensures the quality and integrity of our work.

Relevance: The SOA provides timely research on public policy issues. Our research advances actuarial knowledge while providing critical insights on key policy issues, and thereby provides value to stakeholders and decision makers.

Quantification: The SOA leverages the diverse skill sets of actuaries to provide research and findings that are driven by the best available data and methods. Actuaries use detailed modeling to analyze financial risk and provide distinct insight and quantification. Further, actuarial standards require transparency and the disclosure of the assumptions and analytic approach underlying the work.

Society of Actuaries
475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
www.SOA.org