**Convolutional neural networks**

Convolutional neural networks (CNNs) extract the most important features from an image to classify the image. CNNs use a mixture of convolution layers, pooling layers, activation functions and fully connected layers to classify images. The order and number of these layers is highly variant and depends upon the task at hand.

A convolution layer scans a filter over sections of the image to produce a convolution layer, where the most important information (according to this filter) has been extracted. Filters can be adjusted over time as the network decides what are the most ‘important’ features.

A pooling layer condenses the information from a previous layer – usually a convolution layer – generally taking the maximum value of each section of the image. Other types of pooling are used such as taking the average value of each section of the image.

An activation function sets the point at which information is deemed important enough to ‘activate’ the network, or register an important feature. There exist many activation functions such as ReLu, tanh, and the sigmoid. The final layer(s) are the fully connected layers, where all the information from the previous layers is accumulated to classify the image. A function such as SoftMax can be used to classify outputs as probabilities, such as the probability of a particular image being that of a car, compared to that of a horse.

Neural networks have been widely used for character recognition tasks, as far back as 1984. Chinese characters present a particular problem for neural networks due to the huge amount of classifications (>3700), requiring a huge number of neurons in each convolution layer and a very large fully connected layer at the end of the CNN.

Furthermore, optimising the architecture of CNNs is largely guesswork, rather than having a systematic approach. Too many neurons can either be redundant or can even reduce the accuracy of the overall network. Furthermore the number and order of convolutional and pooling layers can be entirely arbitrary.

One particular advantage of CNNs is that transfer learning can be used, where the network is first trained on another task, and then only the outer layers are trained on the true task. For the task at hand, the network could be pre-trained on a small subset of the data set, or even a different data set such as hand written Latin characters. Previous research has shown that pre-training a network hugely reduces training time as well as improving accuracy.

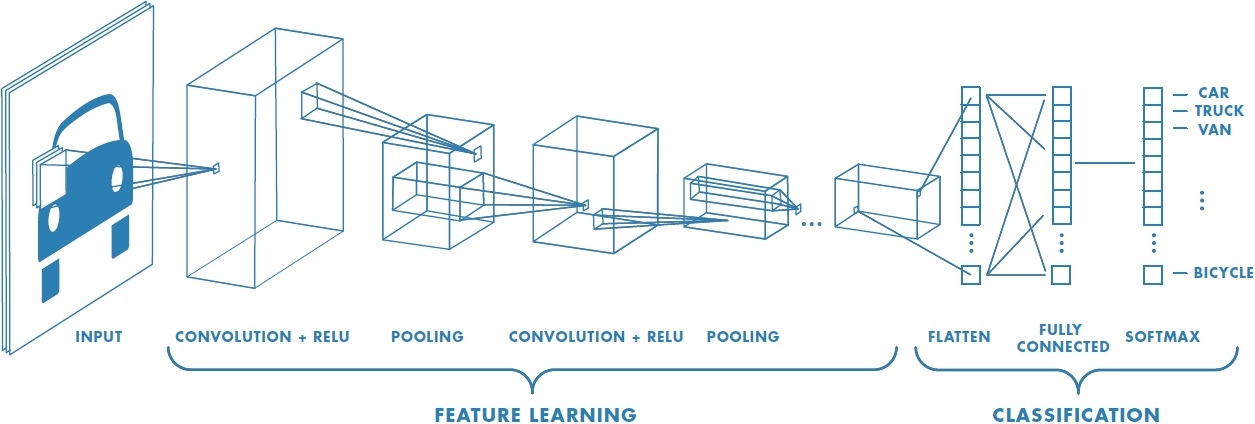


Figure 1: Diagram of a CNN (Convolutional Neural Network, [www.Mathworks.com](http://www.Mathworks.com))

**Wavelet transforms**

Wavelet transforms can be used to transform data – in this case image files – into a set of coefficients classifying spatial frequency. It works like the eigenface method in that it produces an orthonormal basis for the image.

A set of coefficients is generated for each pixel in an image, which can then be used to manipulate the image. Images can be separated into coefficients based on the spatial frequency, with low and medium spatial frequencies corresponding to image content and high frequencies corresponding to noise. These coefficients are generated for each pixel in an image, and then can be used to manipulate the image: reproducing it, removing noise, or compressing the image. Information in the coefficients is statistically concentrated, making it easier to work with.

The coefficient information and the corresponding calculations that can be performed on these such as calculating the entropy, can be used in a machine learning method such as CNNs or random forest to classify data.

One difficulty with wavelet transforms is that there are many different functions that can be used to transform the data. Finding an appropriate function is another case of guesswork.