Project 1 Conventional Neural Network Learning for Dairy Cows Teat-End Condition Classification

Juju Ren
Yeshiva University Computer Vision AIM 5014
Jren@mail.yu.edu

Abstract

The ability of artificial intelligence to close the gap between human and machine skills has significantly increased. Researchers and hobbyists alike work on various facets of the area to achieve spectacular results. The field of computer vision is just one of many such disciplines.

The goal of this field is to give machines the ability to see and understand the world similarly to humans do, and to use that understanding for a variety of tasks like image and video recognition, image analysis and classification, media recreation, recommendation systems, natural language processing, etc. The improvements in Computer Vision using Deep Learning have been built and developed through time, mostly thanks to one specific algorithm: a Convolutional Neural Network.

1. Introduction

1.1. Convolutional Neural Network

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning method that can take in an input image, give various elements and objects in the image importance (learnable weights and biases), and be able to distinguish between them. Comparatively speaking, a ConvNet requires substantially less pre-processing than other classification techniques. ConvNets have the capacity to learn these filters and properties, whereas in primitive techniques filters are hand-engineered. A ConvNet's architecture was influenced by how the Visual Cortex is organized and is similar to the connectivity network of neurons in the human brain. Only in this constrained area of the visual field, known as the Receptive Field, do individual neurons react to stimuli. The entire visual field is covered by a series of such fields that overlap. A multi-layer neural network called CNN has multiple Convolutional layers that alternate with subsampling layers, and each layer is made up of numerous independent neurons. Each neuron's input in the convolutional layer, also known as feature extraction, is coupled to the local receptive field of the preceding layer. The convolutional layer has numerous distinct binary feature maps, and each feature map extracts a single feature. The weights of the same feature map, or the same convolution kernel, are reused when extracting a range of distinct features. Similar to this, various feature maps employ various convolution kernels. For the preservation of local characteristics, the convolutional layer is useless. Yes, the retrieved characteristics are rotation and translation invariant.

The feature mapping layer, also known as the subsampling layer, is in charge of sub-sampling the features produced by the convolution layer in order to give the retrieved features some degree of invariance. It was merely a straightforward scaling mapping in subsampling, with a manageable amount of neurons to train and a straightforward calculation. The number of final output nodes equals the number of classification objectives. There are typically multiple fully linked layers at the end of the CNN. To get the CNN's output as near to the original label as possible, training is done.

Remember that all of the neurons in the layer before receive input from each neuron in the network via connected channels. The output vector from the preceding layer is multiplied by the weighted sum of all the weights at each of these connections to create the input. The activation function receives the weighted sum and produces the output for a specific neuron as well as the input for the following layer. Each layer experiences this forward propagation until the data reaches the output layer.

Through the use of pertinent filters, a ConvNet may effectively capture the spatial and temporal dependencies in a picture. Because there are fewer factors to consider and the weights can be reused, the architecture provides a better fitting to the picture data set. In other words, the network may be trained to better comprehend the level of complexity in the image.

1.2. Case introduction

In this case we are studying to build a CNN model to train Cow-Teat data. The goal for this project is to have the model accuracy in images(cow teat) classification reach to a high level. In the project, we have two data sets. One is ”training data”, the other is ”testing data”. In the training data folder, we have the images labeled with four different scores. The score is labeled based on the teat-feature. There are 1149 images classified in 4 groups in the training data, and 380 unlabeled images in the testing data. To test the model training accuracy we need to evaluate the result using the developed software.

2. Related Work

The goal of image classification is to relate each image to a certain set of class labels. A set of pre-labeled training data is provided to a machine learning algorithm in this supervised learning challenge. The goal of this method is to identify unlabeled photos using the visual characteristics included in the training images that correspond to each label.

2.1. CNN and deep learning

The fields of machine learning and computer vision and speech recognition mainly neglected neural nets and backpropagation in the late 1990s, whereas the machine learning community embraced them. It was often believed that learning practical, multistage feature extractors without much background knowledge was impossible. It was believed, in instance, that simple gradient descent would become stuck in suboptimal local minima, or weight configurations for which no little modification could lower the average error.

CNN offers a specific kind of deep, feedforward network that was significantly simpler to train and generalized much better than networks with full connectivity between neighboring layers.It has recently been widely adopted by the computer vision community.

supervised learning is the most commom form of machine learning, and it is easy to visualize. The method known as stochastic gradient descent is what most professionals utilize in practice (SGD). This involves displaying the input vector for a few examples, calculating the outputs and errors, calculating the average gradient for those cases, and then modifying the weights accordingly. Until the average of the objective function stops declining, the process is repeated for numerous tiny sets of training set samples. Because each tiny group of samples provides a noisy estimate of the average gradient over all examples, it is known as stochastic. When compared to far more complex optimization approaches, this straightforward method frequently yields a good set of weights surprisingly quickly18. Using a different set of examples known as a test set, the system's performance is evaluated following training. This is done to assess the machine's capacity for generalization, or its capacity to generate logical responses to novel inputs that it has never encountered during training.[1]

2.2. Very deep CNN for large scale image study

A fixed-size RGB image with dimensions of 224 224 is used as the ConvNets' training input. The only preprocessing we perform is subtracting from each pixel the mean RGB value calculated on the training set. The picture is run through a series of convolutional (conv.) To increase the accuracy of our result, one important part to build the model is the depth of the model. It was established that representation depth increases classification accuracy and that a typical ConvNet architecture with much higher depth can achieve cutting-edge performance on the ImageNet challenge dataset.[2]

2.3. Transductive Learning

By modifying SCTL model with GoogLeNet, the model accuracy increases from 61.8 to 77.6 percentage.By employing transductive learning, the paper minimize the divergence between training and test data with a categorical MMD loss.[3]

3. Methods

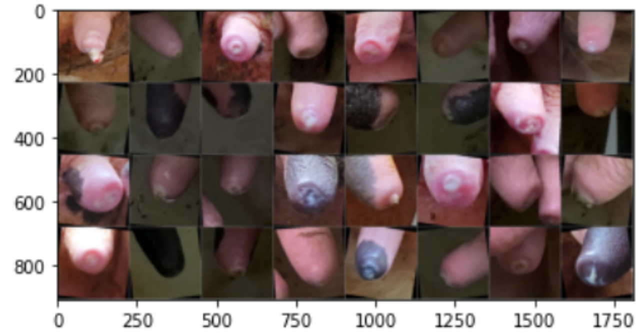
3.1. Motivation

The objective of the paper is to create a fully automated deep learning model using a convolutional neural network (CNN) that can recognize several categories of dairy cow teat-end conditions with accuracy. Finding hyperkeratosis in the vicinity of the teat end is the utilitarian objective. The teat-end image classification task is the subject of our research, and by combining multiple layers with the CNN method, we hope to increase the classification accuracy.

3.2. Datasets

We have required all the data sets in folders. By studying the cow teat end, we need to subtract the feature of the cow teat images to build the model. Before building the model, soring the data is a very import step. The test dataset consisted of 380 teat pictures, or roughly 70 cows. In order to fairly compare various methods, we divided the dataset into

Figure 1. dataset



training (75 percentage, 1149 photos, about 288 cows) and test (25 percentage, 380 images, approximately 70 cows). Based on the test dataset, all findings are presented.

Table 1. . Statistics of training and test data.

2*Label	Train Data		Test Data	
	Number	Percentage	Number	Percentage
Score 1	450	39.2	149	39.2
Score 2	491	42.7	163	42.9
Score 3	187	16.3	62	16.3
Score 4	21	1.8	6	1.6

In the cow-teat study, we split the training data into two groups: training data and validation data. We separate the data randomly and assigned 85 percentage data into training data set, and 15 percentage of data into validation data set.

A robust model requires a lot of data, but there may not be enough data in actual operation, so there is a data enhancement technology, such as shooting a type of picture from different angles, shooting under different light levels, etc., to increase this type of image. diversity.

(1) Flip: Flip (2) Rotation: Rotation (3) Scale: scaling, Resize function (4) Crop: Crop (5) Add noise to the original data Image Transform is usually an indispensable part, which can be used for image preprocessing, data enhancement, etc. Here in the model, we firstly resized all the images into (224, 224) sizes, and we did a horizontal flip with p =0.5, also a 20 degrees of random rotation. Then we transformed the image data into tensor and also normalized the data.

3.3. Build Model

nn.Module is the parent class of all network layers. The linear layer and convolutional layer provided by pytorch are also integrated from this class. nn.module provides a large number of ready-made computing modules, such as the following: Linear, Relu, Sigmoid, Con2d, ConvTransposed2d, Dropout, etc.

When we define our own network layer, we should also inherit this class and implement the feedforward function forward. We have built a model by using the nn.Conv2d, nn.BatchNorm2d, nn.ReLU, nn.Linear, nn.MaxPool2d. Inputting image size of (3, 224, 224), firstly we change the data from 3 channels into 20 channels output, with a filter size of 3, stride size of 1, and padding size of 1. After the convolutional layer, the image size output is (12, 224, 224). the pool layer changed the image size to (12, 112, 112) The second convolutional layer changed the image size to (20, 112, 112) with a filter size of 3, stride size of 1, and padding size of 1. The third convolutional layer changed the image size to (32, 112, 112) with a filter size of 3, stride size of 1, and padding size of 1. finally with an input feature of 32*32*112, the data be transformed into one dimension and the linear filter changed the form into the output classes number =4.

4. Results

After epochs = 100 training, we have plotted the loss and accurate charts to compare the change tendencies between train data and valid data. And finally, using the saved model to predict the test data, we get the testing result. After saving the result into csv file and compare with the developed software, we get the prediction accuracy of 42.6316To improve the result, I am thinking of methods of implementing more features in the steps of transforming data also imput more layers on module building. As we can see from the charts below, the training accuracy is developing positively while the testing accuracy is not.

Figure 2. Loss chart of train and valid data

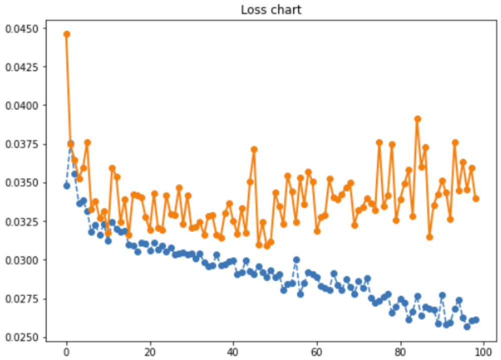
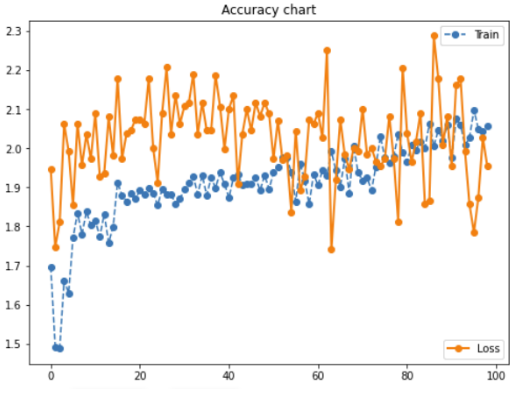


Figure 3. Accuracy chart of train and valid data

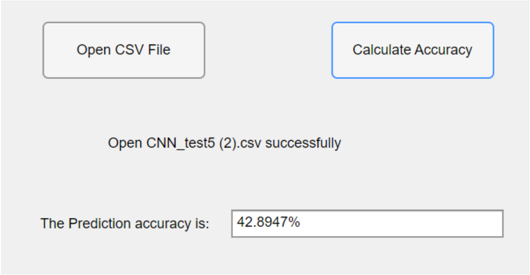


Also we have been tracking the accuracy for all classes to compare the result. We can see that the accuracy declines as the classes number increases, and some reason for this result is because of the low volume of data set in some classes.

5. Discussion

By building the module from scratch, we have noticed there are many factors can effect the end result. Even for

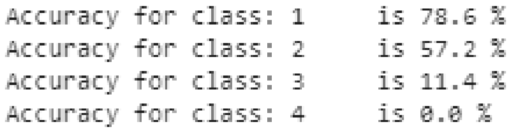
Figure 4. Accuracy test by developed software



Open CNN_test5 (2).csv successfully

The Prediction accuracy is: 42.8947%

Figure 5. Accuracy for 4 different classes



```
Accuracy for class: 1     is 78.6 %
Accuracy for class: 2     is 57.2 %
Accuracy for class: 3     is 11.4 %
Accuracy for class: 4     is 0.0 %
```

the same model, the epoch steps can lead result to under fitting or over fitting. To identify over fitting visually by plotting your loss and accuracy metrics and seeing where the performance metrics converge for both data sets.

6. Conclusion

In this article, we provide a CNN learning model for classifying tail-end images. As a starting point, we suggest a separation loss to widen the distinctions between various groups. We then use adjustment learning to refine the network and get reliable labels for the test data. Last, CNN learning is used to reduce the difference between training and test data loss. Such a model can be helpful with medical consultant with animal farm or vet, which can be useful for developing a more complicated model for classification practice.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 2

[2] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[3] Y. Zhang, I. R. Porter, M. Wieland, and P. S. Basran. Separable confident transductive learning for dairy cows teat-end condition classification. *Animals*, 12(7):886, 2022. 2