Convolution Neural Network Image Classification of Dairy Cow Teats for Health

Aishwarya Deshmane
Yeshiva University
adeshman@mail.yu.edu

Abstract

The well-being of dairy cows is crucial to the production of high-quality milk. This research report focuses on the teat-health evaluation, out of all the other cow health examinations. 1149 images are used to train a convolution neural network, which divides the data of cow teats into four groups. The level of hyperkeratosis in the teats determines which category they fall into. On a different, unseen test dataset consisting of 380 images, the suggested model obtained an accuracy of 62.37%. This research shows that training the model with small batch size, adding residual blocks, and increasing the number of convolution layers all improve accuracy.

1. Introduction

Mastitis is one of the three primary diseases influencing the health of cows and, consequently, the profitability of dairy farmers. It is also one of the most common diseases that afflict cows. The other two are lameness and fertility[5]. Thus, monitoring cow health is crucial, but doing so requires a lot of human labor and effort. The study presented in this paper addresses this issue. This study demonstrates how machine learning neural network techniques, coupled with picture input to the model, can automate the task of assessing cow health teats.

Studies have shown that poor teat health might reduce milk output and increase the amount of time that milk is collected. This serves as a motivator to maintain control over the teat condition. In an investigation, teats were placed in chemically produced chapping conditions and milked with the aid of autonomous take-off clusters. Researchers found that milk yield dropped by 3.6%–8.5% and milking time rose by 1.3-2 minutes, or 26%–51%, when teat health declined. Milk production rose and milking time reduced as the teats healed[1].

The goal of this paper is to predict the levels of hyperkeratosis into the four distinct classes—scores 1, 2, 3, and 4 that G. Mein used[4][2]. Score 1 or N: normal with no callosity ring present, Score 2 or S: smooth or slightly rough ring, Score 3 or R: rough ring, Score 4 or VR: very rough ring.

2. Related Work

Deep learning has shown remarkable promise in recent years for a variety of computer vision applications, such as image classification, object detection, image segmentation etc. Convolutional Neural Networks (CNNs) are among the most popular deep learning architectures for image classification. ResNet is one of the most well-liked CNN architectures because of its capacity to address the vanishing gradient issue and enhance deep network performance.

Transfer learning is one method for approaching computer vision and image classification, regardless of the model one uses. By adopting models that have already been built and then adjusted to the problem being solved, this makes use of the power of community improvements. I. Porter et al. attempted this approach for the first time with the cow teat categorization problem [6]. Utilizing the pre-trained GoogleNet [7] model, which was trained on the 1,000-class ImageNet dataset, they were able to identify the photos using the capabilities of convolutional neural networks (CNNs). Porter and colleagues refined this model using the cow teat dataset, utilizing the models pre-existing expertise.

An alternative strategy involves enhancing an extant model through the application of innovative techniques, as demonstrated by Y. Zhang et al. SCTL [8]. Their method involved training seventeen benchmark models on the ImageNet dataset using an enhanced form of transductive learning known as separable confident transductive learning (SCTL). By employing the SCTL approach, they are able to decrease the difference between intra-class differences and enhance the discrimination between the various classes. As a result, they were able to enhance the accuracy of the benchmark ImageNet models by an average of 4.9%; the GoogLeNet [7] model saw the most gain, at 13.4%, while the SqueezeNet [3] model saw the lowest, at 0.8%.

3. Methods

The goal of this paper is to develop Convolution Neural Network(CNN) to classify the cow teat health into 4 categories using cow teat images.

The CNN developed in this work has a total of 20 convolution layers, 5 residual layers, an input image size of (224, 224, 3), and 4 output classes.

There are 5 blocks in this model as shown in Figure 2, and each block consists of 3 convolution layers that follow one another. The input and output channels, padding, stride, and kernel size are present in every layer. The kernel measures 5x5 for the first layer and 3x3 for the other two. The stride and padding are both set to 1. The ReLU activation function in this series comes after the first two convolution layers. The residual convolution layer with kernel size 3x3, stride 1, and padding 0 comes next in this order. The normalizing layer is subsequently applied to these outputs, and thereafter, the max-pooling layer with a kernel and stride of 2 is sent through. The image dimension will reduce by half as a result. Ultimately, the output is flattened and transferred to the fully connected layer.

In the forward method, we defined the sequence of the

Layer(type)	Output Shape	Param #
Conv2d-1	[-1, 32, 222, 222]	2,432
ReLU-2	[-1, 32, 222, 222]	0
Conv2d-3	[-1, 32, 222, 222]	9,248
ReLU-4	[-1, 32, 222, 222]	0
Conv2d-5	[-1, 32, 222, 222]	9,248
Conv2d-6	[-1, 32, 222, 222]	896
BatchNorm2d-7	[-1, 32, 222, 222]	64
MaxPool2d-8	[-1, 32, 111, 111]	0
Conv2d-9	[-1, 64, 109, 109]	51,264
ReLU-10	[-1, 64, 109, 109]	0
Conv2d-11	[-1, 64, 109, 109]	36,928
ReLU-12	[-1, 64, 109, 109]	0
Conv2d-13	[-1, 64, 109, 109]	36,928
Conv2d-14	[-1, 64, 109, 109]	18,496
BatchNorm2d-15	[-1, 64, 109, 109]	128
MaxPool2d-16	[-1, 64, 54, 54]	0
Conv2d-17	[-1, 128, 52, 52]	204,928
ReLU-18	[-1, 128, 52, 52]	0
Conv2d-19	[-1, 128, 52, 52]	147,584
ReLU-20	[-1, 128, 52, 52]	0
Conv2d-21	[-1, 128, 52, 52]	147,584
Conv2d-22	[-1, 128, 52, 52]	73,856
BatchNorm2d-23	[-1, 128, 52, 52]	256
MaxPool2d-24	[-1, 128, 26, 26]	0
Conv2d-25	[-1, 256, 24, 24]	819,456
ReLU-26	[-1, 256, 24, 24]	0
Conv2d-27	[-1, 256, 24, 24]	590,080
ReLU-28	[-1, 256, 24, 24]	0
Conv2d-29	[-1, 256, 24, 24]	590,080
Conv2d-30	[-1, 256, 24, 24]	295,168
BatchNorm2d-31	[-1, 256, 24, 24]	512
MaxPool2d-32	[-1, 256, 12, 12]	0
Conv2d-33	[-1, 512, 10, 10]	3,277,312
ReLU-34	[-1, 512, 10, 10]	0
Conv2d-35	[-1, 512, 10, 10]	2,359,808
ReLU-36	[-1, 512, 10, 10]	0
Conv2d-37	[-1, 512, 10, 10]	2,359,808
Conv2d-38	[-1, 512, 10, 10]	1,180,160
BatchNorm2d-39	[-1, 512, 10, 10]	1,024
MaxPool2d-40	[-1, 512, 5, 5]	0
Linear-41	[-1, 512]	6,554,112
Dropout-42	[-1, 512]	0
ReLU-43	[-1, 512]	0
Linear-44	[-1, 4]	2,052

Figure 1. Summary of the implemented model. Total number of trainable parameters is 18.77Mn

layers taking into account all the blocks in the model. The output from this is reshaped and passed as an input to the fully connected layer.

There are 18,769,412 trainable parameters in all for this neural network model.

Let's understand what is the function of each layer used here:

3.1. Convolution

The convolution layer is the basic building element of a CNN. Convolution is the process of multiplying and adding the filters and the input data. With the input image, it applies the convolution process using the kernel/filter, padding, and stride. In doing so, the convolved output is created, and the patterns in the input data are captured.

3.2. Relu

This layer applies the rectified linear unit activation function, setting all the negative values to zero and positives values remain unchanged. It introduces non-linearity to the model.

ReLU(x) = max(0,x).

3.3. Batch Normalization

Normalizes the input images which improves the performance and avoids the overfitting.

3.4. MaxPooling

Pools the maximum value within the pooling window while sliding through the image. This reduces the image size but gives the image with important features.

3.5. Fully Connected layer

Transforms the input to linear and flattened output.

3.6. Dropout

This is a regularization layer that prevents overfitting of data while training. Dropout randomly sets a fraction of the neurons output to zero, effectively dropping out some units.

3.7. Residual block

While training the Neural network with gradient-based learning methods, a vanishing gradient problem may arise and prevent the weight from changing its values. To avoid the problem, adding the residual block to the network helps. Here X is the input to the block and F(X) is the residual mapping to be learned. F(X) is the shortcut or the skip connection that allows the model to skip the layers and help the vanishing gradient problem. This allows direct back-propagation to previous layers making it easy to optimize.

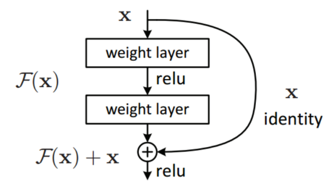


Figure 3. Residual block

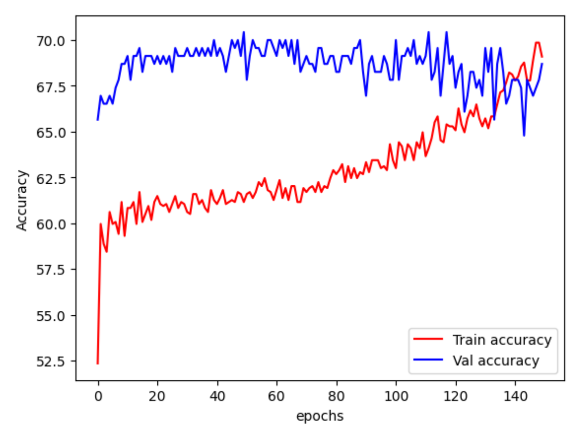
3.8. Hyperparameters

CrossEntropy Loss, a popular loss function for classification model optimization, is applied here.

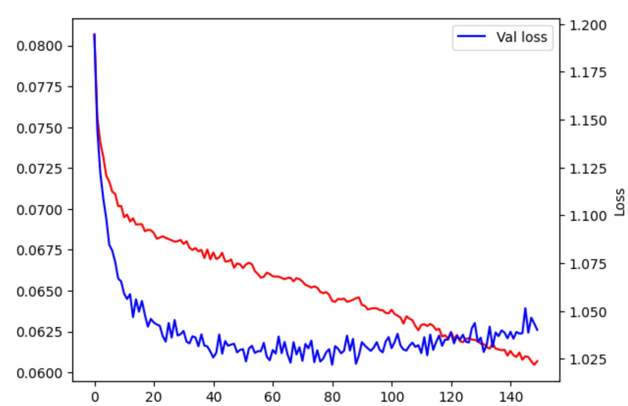
In this case, stochastic gradient descent(SGD) is the optimizer user. I got better outcomes with SGD than I did from Adam.

The model with the highest accuracy has a train batch size of 16 and 150 epochs. An increase in epochs improved training accuracy but lowered validation and test accuracy because of overfitting.

4. Results



(a) Train vs Validation accuracy



(b) Train vs Validation loss

Figure 4. Accuracy and loss

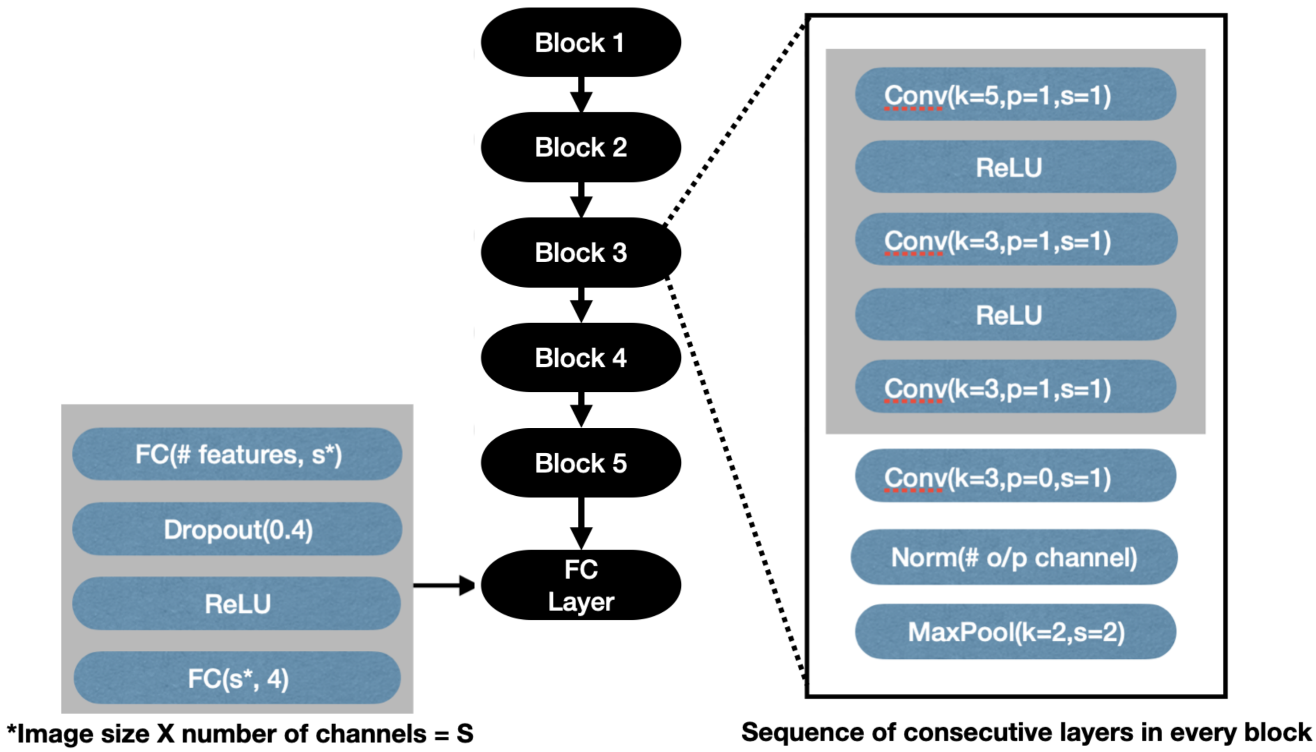


Figure 2. The architecture of the model implemented in this paper.There are 5 blocks and each block is a sequence of 3 consecutive convolution layers with ReLU activation function between them. This is followed with the residual convolution layer, batch normalization and maxpooling. The output from block 5 is passed to the fully connect layer. Fully connected layer consists a sequence of 4 layers fully connected layer, dropout of 0.4, ReLU and fully connected layer with 4 categories as output.

As you can see in Figure 5, after experimenting with several models and hyperparameter combinations, Model 5 produced the best results for me. With a train batch size of 16, or 150 fine-tuned epochs, this model yielded an accuracy of 62.37% on the unseen dataset. Increasing the number of epochs results into overfitting.

Figure 4 showcase the accuracy and loss plots for training vs validation set for Model 5.

4.1. Datasets

This dataset comes from Y Zhang[1] on Github, and I used it. 1149 training images and 380 test images in 4 distinct categories make up the dataset. The dataset distribution table displays the class distribution.

The majority of the images in the dataset are from the score 2 class, followed by images from the score 1 class. The dataset's least contributing images are those with a score of 4. When the model is tested on test data, it detects class 1, 2, and 3 images effectively, but has trouble identifying class 4 images. It's because there wasn't enough data to train the model effectively on class 4 images. The test dataset's model accuracy can be only evaluated using a Matlab software developed by Y Zhang. The train data is split into train and validation set with 80%-20% since there is no way to validate the test data. This resulted into 919 images into train set and 230 images in validation set. The images were normalized and then resized to 224*224. It is observed that, normalizing the images enhances the efficiency and the performance of the models.

Table 1. Data distribution

Class	Count	Percent
1	450	39.16%
2	491	42.73%
3	187	16.28%
4	21	1.83%
Total	**1149**	**100.00%**

5. Discussion

The number of convolution layers, filters, residuals, batch normalization, and dropout seams in the model all have an impact on its accuracy, as we have shown in earlier sections. In order to determine whether the relationship holds true as the number increases, it will be fascinating to compare even more convolutions, filters, and residuals in future work. Play with the hyper parameters and determine

Model	Conv layers	Parameters	Optimizer	Train Batch size	Epoch	Test_acc
Model 1	4	6 Mn	SGD	64	300	20.26%
Model 2	20	18.77 Mn	Adam	64	300	52.39%
Model 3	24	49.97 Mn	SGD	64	300	61.58%
Model 4	20	18.77 Mn	SGD	64	300	61.58%
Model 5	**20**	**18.77 Mn**	**SGD**	**16**	**150**	**62.37%**
Model 6	20	18.77 Mn	SGD	64	400	62.11%

Figure 5. Model results

if the accuracy converges to a certain number.

6. Conclusion

In conclusion, this study explores the vital area of dairy cow health, paying special attention to teat health. In order to overcome the difficulty of manual assessment, the Convolutional Neural Network (CNN) that was developed shows encouraging results in classifying teat health into four different classes. The results highlight how crucial it is to train the model with a small batch size, include residual blocks, and add more convolution layers in order to improve accuracy. The 20 convolution layers in the CNN that is being presented show how machine learning can be used to automate the evaluation of cow teat health.

The built model gave an accuracy of 62.37% for the unseen dataset. In order to further maximize model accuracy, the discussion recommends directions for future study and encourages the investigation of extra convolution layers, filters, and residuals. In summary, this work provides important new understandings of the relationship between machine learning and the health of dairy cows, laying the groundwork for future efforts to improve and extend the CNN model. The potential for automating cow health assessments holds promise for improving the general productivity and efficiency of dairy farming practices as long as technological advancements persist.

References

[1] Tanya Eadie. Dairy cow teat condition scoring, 2020. 1

[2] Alejandro Carmelo Parra Garcia. Convolutional neural networks and residual connections for cow teat image classification. *arXiv preprint arXiv:1409.1556*, 1, 2014. 1

[3] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 2

[4] GA Mein, F Neijenhuis, WF Morgan, DJ Reinemann, JE Hillerton, JR Baines, I Ohnstad, MD Rasmussen, L Timms, JS Britt, et al. Evaluation of bovine teat condition in commercial dairy herds: 1. non-infectious factors. In *Proceedings of the 2nd International symposium on mastitis and milk quality*, pages 347–351. Citeseer, 2001. 1

[5] D O'Rourke. Nutrition and udder health in dairy cows: a review. *Irish Veterinary Journal*, 62(4):1–6, 2009. 1

[6] IR Porter, Matthias Wieland, and PS Basran. Feasibility of the use of deep learning classification of teat-end condition in holstein cattle. *Journal of Dairy Science*, 104(4):4529–4536, 2021. 2

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[8] Youshan Zhang, Ian R Porter, Matthias Wieland, and Parminder S Basran. Separable confident transductive learning for dairy cows teat-end condition classification. *Animals*, 12(7):886, 2022. 2