Concordia University

Mask Detection Using A Neural Network

GitHub Repository

Team NS_04

Compliance Specialist: Maxime Johnson (40081684)
Training Specialist: Visweshwaran Balasubramanian (40071701)
Evaluation Specialist: Matthew Greco (40136180)
Data Specialist: Hayden Ross (40131155)

Artificial Intelligence (COMP 472)
Dr. René Witte
2022/06/08

DATA PREPROCESSING

For this project, it was extremely important that we obtain hundreds of photos of people wearing each type of mask (or none at all). This presents a unique challenge as to produce good results these photos need to have a single person in them, and ideally, the people's faces wouldn't be obstructed in any way (except for the masks of course), although minor obstructions are included. In order to prepare our model for unexpected data a large variety of images representing a diversity in race, gender, and headwear were included.

High definition photos with easy visibility were chosen, as we want the photos to be visible. The ideal photo included the subject's full face and mask, when applicable, from any angle. Some imperfect photos are included in the training data to enable a versatile AI: for example, a photo where someone's chin falls out of sight. Photos where someone's mask was improperly on were also not included, since they wouldn't fit in any class. Similarly, masks that seem to be made with cloth but have filters were also excluded. The photos in each class are above the minimum size of 100 by 100 pixels.

Ideally we should have roughly the same number of photos for each class to avoid any bias. This bias can lead to poorer results. To compensate for the lack of N95 usage and therefore accessible photos within a single dataset, they had to be searched for independently. Due to time constraints, there is still an imbalance. Currently, there are 441 photos in the cloth class, 283 in the N95 class, 373 in the maskless class, and 421 in the surgical class.

The data used in this project was mostly found from a dataset on the website "Humans in the Loop". This dataset included over 6000 photos of varying quality, poses, masks and subjects.

In order to sort the data, a team member created a python script that will show the user a photo, take the user's decision on what type of photo it is, save the photo in the appropriate folder, and then show the next image in the folder. This is semi-manual, and naturally still takes a lot of time. There is no quicker way to manually sort through data.

In preprocessing, there is a resizing of every photo to 100 by 100 pixels and a normalization of the color values, this is applied to both the train and test data sets [5].

CNN MODEL ARCHITECTURE

The CNN architecture developed for identifying masks in images, can be seen in the diagram below:

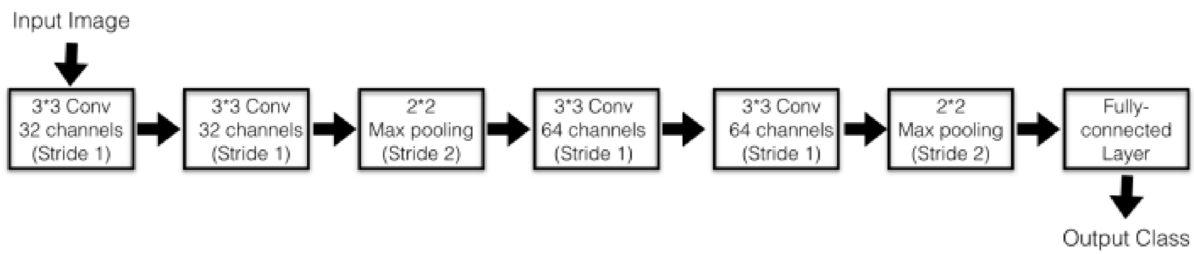


Fig. 1 Neural Network Architecture

As can be seen, there are 4 convolutional layers, 2 max-pooling layers and one fully connected layer. This setup is similar to the one developed in Lab 7 (CIFAR10 dataset classification) [1]. We also sequentially order the layers in the network to create a convolution + leaky ReLU (with non-zero slope for <0 activation output) + pooling sequence. Also to be noted, we use 1 pixel of padding on the sides to prevent data loss from the perimeter of the image.

The convolutional layers utilize a 3x3 kernel and 32 channels for finding features. The first convolutional layer has 3 channel input (image in RGB colorspace) and 32 channel output, leading into the next convolutional layer [2].

After two rounds of extrapolating features, we use 2x2 max-pooling with strides of 2 pixels, to select the maximum element from the region of the feature map covered by the convolution filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map [3].

We repeat this process again, but this time using 64 channels for the next two convolutional layers. This is done as the first few layers only extract low level features from the image, like lines and edges. Further layering extracts higher-level features and makes detection easier.

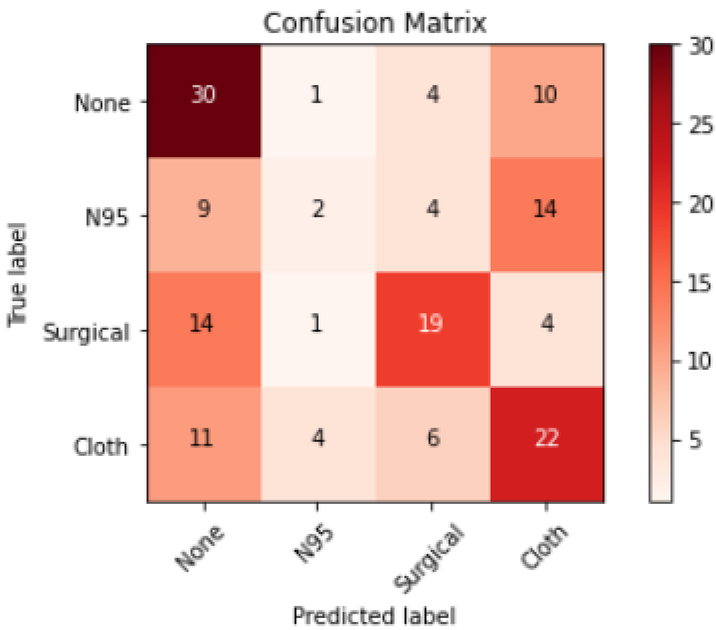
When we do max-pooling the second time here, we extract the most prominent high-level features. We then use these features in the fully connected feature to arrive upon one of the classes being detected [4]. The CNN architecture and its parameters could be changed while trying to achieve good detection metrics.

With the CNN being defined, the team decided that training should be done over 4 epochs with a learning rate of 0.0003. We loop through the number of epochs while passing CNN model output and true labels to the CrossEntropyLoss function. Then we perform back propagation to calculate gradients and then an optimized training step.

EVALUATION

Shown below is the generated confusion matrix from training the neural network.

Fig.2 Generated confusion matrix from test training



Metric	Percentage
Precision	45%
Recall	47%
Accuracy	47%
F1	0.44

The accuracy and recall for our CNN are both 47%, the precision is 45% while the f1 measure is 44%. These prediction metrics are all rather low for a CNN but we do have clear ideas on how we can improve our CNN in part two of the project. There isn't much to deduce from our evaluations since they are all low. We have a high number of false negatives leading to a low recall, a high number of false positives leading to low precision which in turn causes a poor f1 measure and accuracy.

The confusion matrix shows that it is quite good at predicting cloth and surgical masks while only predicting n95 once and incorrectly at that. This led to 0 of the n95 masks to be predicted correctly. Therefore we can gain up to 25% accuracy by improving our n95 predictions.

As previously mentioned, our data is slightly skewed since we had trouble gathering enough n95 images. This causes our CNN to nearly never predict n95 since the odds of the image being of that type is less than other types. Therefore, in part 2 we will add enough n95 images to have an even distribution across all mask types to remove the current bias. Additionally, to improve overall prediction we want to zoom in on the subject's face so that our CNN can ignore most of the background noise.

REFERENCES

Image refs:

2021, *Medical Mask Dataset*, Humans in the Loop, https://humansintheloop.org/mask-dataset-download/?submissionGuid=b7e8fcde-7f69-4b8e-b41f-1eb4284c5619

AP Photo/John Hefti, Nov. 11, 2018. https://abc7news.com/n-95-mask-with-valve-kn95-n95-for-smoke/6378125/

Arturo Rey, June 2, 2020, Unsplash, https://unsplash.com/photos/5yP83RhaFGA

Ashkan Forouzani, March 17, 2020, Unsplash, https://unsplash.com/photos/IXSgwfBGnCg https://unsplash.com/photos/I-NIPb-9Njg?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink

Engin Akyurt, April 6, 2020, https://unsplash.com/photos/0zNQ3XpbwOY

Engin Akyurt, April 2, 2020, https://unsplash.com/photos/E9dJTLiizCE

Kate Trifo, March 26, 2020, Unsplash, https://unsplash.com/photos/ZPTduQKik6c

Kay Lau, March 17, 2020, Unsplash, https://unsplash.com/photos/vvgSoo6fvXM

Önder Örtel, March 25, 2021, Unsplash, https://unsplash.com/photos/ddJ18LY8s48?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink https://unsplash.com/photos/ZPTduQKik6c

Code refs:

[1]: Concordia University, COMP 472 Lab 07 Introduction to Deep Learning, https://moodle.concordia.ca/moodle/mod/resource/view.php?id=3136494

[2]: Sabyasachi Sahoo (Aug, 2018), Deciding Optimal Kernel Size for CNN, https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363

[3]: GeeksForGeeks (Jul, 2021), CNN | Introduction to Pooling Layer, https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

[4]: User:Djib2011 (Sept, 2018), Stack Exchange: Intuition behind 2 consecutive convolutional layers, https://datascience.stackexchange.com/questions/37975/what-is-the-intuition-behind-using-2-consecutive-convolutional-filters-in-a-conv

[5] PyTorch (2022), Datasets & DataLoaders, https://pytorch.org/tutorials/beginner/basics/data_tutorial.html

Tech Dose, Resize or Scale an image using python || 5 different types of image resize, Youtube, https://www.youtube.com/watch?v=cWHW9MnX_F4&t=1s December 6th, 2018

“Confusion Matrix¶.” *Scikit*, https://scikit-learn.org/0.18/auto_examples/model_selection/plot_confusion_matrix.html.

Mandy. “Create a Confusion Matrix for Neural Network Predictions.” *YouTube*, Deeplizard, 9 July 2020, https://www.youtube.com/watch?v=VYGVOwlT4oE&list=PLZbbT5o_s2xrwRnXk_yCPtnqqo4_u2YGL&index=8.