Computational Models for Chest X-Ray COVID-19 detection

Oliver Li
gaotang@umich.edu

Jack Yu
yuwenjin@umich.edu

Christophe Shen
chrisslt@umich.edu

1. Introduction

The COVID-19 pandemic has increased in scale and dimension since its outbreak in 2019 [7]. Though initial symptoms might be mild, severe health concerns will appear in later phases, making early detection of this illness emergent. Traditionally, physicians relied on anecdotal experience to determine symptoms from images of CT scans or through real-time reverse transcription-polymerase chain reaction (RT-PCR) tests [5], which oftentimes require high manual costs and are prone to subjective misjudgments.

Additionally, researchers have shown that PCR tests have the issue of being time-consuming and having low sensitivity [14]. To address these issues, recent works propose to leverage computational models and algorithms to detect this illness from CT and chest X-ray scans [13]. Past works attempt at developing models to analyze CT scans including various neural network models and different image pre-processing techniques. However, the primary models employed in the past are variant convolutional neural networks (CNN) with simple augmentations. Contrary to the past works that use an advanced or less parametrized network, we believe that some simple modifications to naive CNNs can contribute to a decent performance. Furthermore, we explore the possibility of leveraging a contrastive learning framework to learn the classification task. We summarize our main contributions as follows:

- We implement everything from scratch (build dataloaders, training framework, etc.) and provide a ready-to-use code source for PyTorch users.

- We design a CustomCNN model and compare it against several baselines. Our finding suggests that simple modifications to the naive CNN model guarantee a competitive performance.

- We propose a novel contrastive learning framework for this supervised learning task. Our proposed framework achieves comparable results to the prior SOTA approaches.

2. Related Work

One recent work adopts white balance as the image processing step to deal with medical images' low lighting attribute that normally results in a darker image compared to the ground truth [14]. This work also constructs a depthwise separable convolutional neural network (DSCNN) that achieves 96.43% accuracy for multi-class classification. Another line of work uses a combination of the Fast Fourier Transform technique and Gabor filter to process the chest CT scans and achieves an overall 95.99% sensitivity [4]. Furthermore, image processing methods used for MRI images (adaptive thresholding and Shannon's entropy) are also considered given the two image types' similarities in overall black-and-white color and lack of contrast in some areas [8]. However, we would like to raise the question that if it is necessary to use more advanced and complicated models. Moreover, current methods are not designed for leveraging data augmentations despite the advanced pre-processing techniques available. In our work, we would like to answer the question and attempt at building a framework that can be utilizing the full potential of data augmentation in our settings.

3. Methods

This project creates a custom convolutional neural network (CNN) and a contrastive learning network in combination with image pre-processing approaches to classify the lung conditions (Covid, Normal, Virus) from chest X-ray images.

Inspired by related works, three pre-processing methods are adopted to train models in this research. This includes Canny Edge detection [6], adaptive thresholding, and Shannon's entropy [12]. All three methods are related to edge detection and can potentially identify features that convey information about COVID-19 positive state. Additionally, every raw input image is converted to gray-scale and resized to 224x224.

3.1. Custom CNN

The custom CNN is inspired by the Mini-VGG coded in Homework 5 [11] and adopts several adjustments. First,

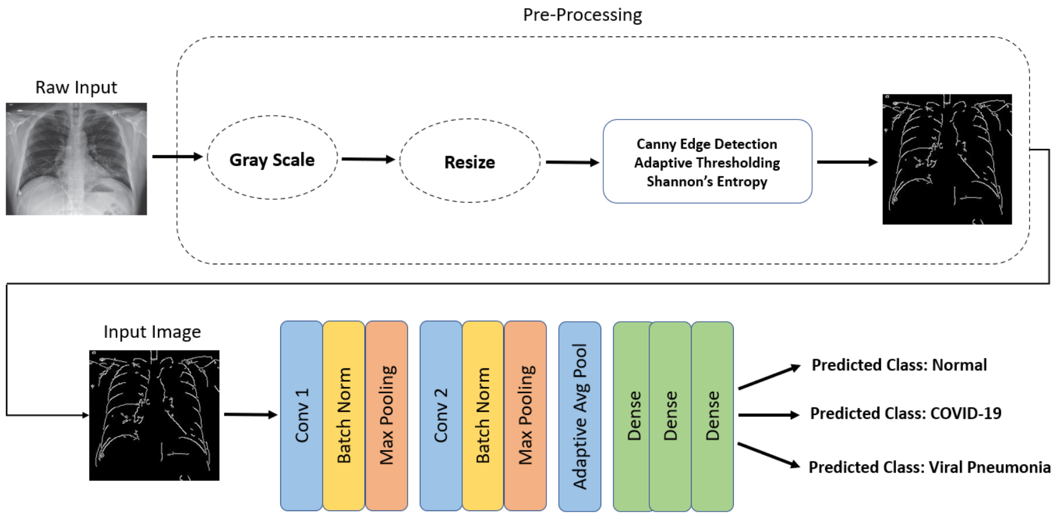

Figure 1. Training methodology of the proposed custom CNN.

```
CUSTOM_CNN(
  (features): Sequential(
    (0): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
    (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(5, 5))
  (classifier): Sequential(
    (0): Linear(in_features=1600, out_features=512, bias=True)
    (1): ReLU(inplace=True)
    (2): Linear(in_features=512, out_features=256, bias=True)
    (3): ReLU(inplace=True)
    (4): Linear(in_features=256, out_features=128, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=128, out_features=64, bias=True)
  )
)
```

Figure 2. Custom CNN architecture with custom convolutional and fully connected layers.

the convolutional layers are shortened to two layers without padding. The reasoning is the X-ray image is small and key lung features are centered in the image, making padding on the edge not necessary. Second, the input channels and output channels are generally smaller in size compared to Mini-VGG with the intention to decrease the number of training parameters for more efficient model training. Third, one more fully connected layer is added to accommodate the smaller number of neuron outputs from the convolutional layers. Fourth, the fully connected layers do not have dropouts since the hypothesis for detecting positive COVID results is that small regions in the picture represent the positive status, and dropping some features learned may omit these regions and lead to inaccurate model learning. The overall methodology is depicted in Figure 1 and the specific network parameters are depicted in Figure 2.

3.2. Supervised Contrastive Learning

In addition to the customized CNN model, we propose to employ a contrastive learning approach in our setting to leverage the efficacy of the popular data augmentation techniques. On the contrary to a variety of self-supervised contrastive learning methods that experiment in a "labelless" setting, we can train a backbone model that extracts useful features by leveraging the ground truth information.

We propose a two-stage approach to learning robust encoders and classifiers. In stage 1, we train a robust encoder that provides condensed and meaningful vector representations of the original inputs. In stage 2, we feed these extracted features to train a final linear classifier for predictions. Figure 3 provides a graphical overview of our algorithm.

3.2.1 Stage 1: Encoding Stage

The key intuition behind using the idea of contrastive learning for the supervised problem builds on the empirical observations that supervision provides additional help with learning the encoder in addition to augmentations only. We adopt the SupLoss from [10] and we have the following objective to be minimized in Stage 1:

$$\mathcal{L}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp z_i \cdot z_p / \tau}{\sum_{a \in A(i)} \exp z_i \cdot z_a / \tau} \quad (1)$$

Where $P(I) = \{p \in A(i) \colon \tilde{y}_p = \tilde{y}_i\}$ refers the set of indices of all positives in the minibatch distinct from i with $P(i)$ representing its identity. We use the simple dot products to denote similarity in the latent space.

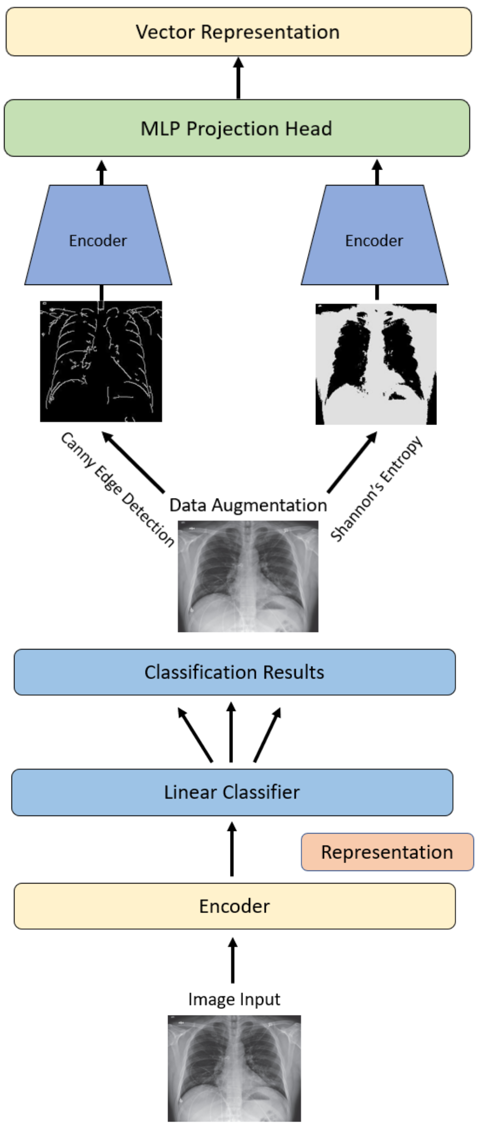

Figure 3. 2 stages supervised contrastive learning methodology.

Similar to popular contrastive learning frameworks, our Stage 1 algorithm consists of the following modules:

- *Data Augmentation* module, *Aug*$(\cdot)$. For each sample input x, we generate two augmentations based on Canny Edge detection, adaptive thresholding, or Shannon's entropy.
- *Encoder Network*, *Enc*$(\cdot)$. For each augmented input $\tilde{x}$, the encoder network maps it to a representation vector in the latent space $r \in \mathbb{R}^{D_{\text{latent}}}$.
- *Projection Network*, *Proj* $(\cdot)$. After obtaining r, we feed it into a multi-layer perceptron with a single hidden layer of size D_{latent} and minimize the objective function 1.

3.2.2 Stage 2: Classifying Stage

After learning the robust feature extractor, we train a linear classifier for end-to-end predictions. We use raw images for input in Stage 2 and freeze the encoder learned from Stage 1. For each image, we use the encoder to generate a representation vector and feed it into the linear classifier. We use the standard empirical risk minimization objectives and the cross-entropy loss for training.

3.2.3 Algorithm

For better presentation, we summarize our algorithm as follows:

Algorithm 1 Supervised Contrastive Learning

Input: Training Dataset D, temperature τ, number of epochs in Stage 1 T

Stage 1: Encoding Stage

1. Generate Augmented Dataset $\tilde{D}$.
2. Train the *Encoder Network* through Optimize $\mathcal{L}^{sup}$ 1 for T epochs.

Stage 2: Classifying Stage

3. For each input x in the raw training D, generate its representation vector through the *Encoder Network* from Stage 1 $r = Enc(x)$.
4. Feed the encoded representation vector r into the linear classifier and train until convergence.

3.3. Evaluation

To compare the two proposed models' results, two existing network structures: Mini-VGG adopted from Visual Geometry Group network [15] and Residual Network [9] (Resnet18) are trained. A grid search is then conducted based on the previously described models and preprocessing methods, rating by testing accuracy on the existing dataset. Each model has 10 iterations of training with average accuracy for each classification class recorded and a receiver operating characteristic (ROC) curve for each class saved.

4. Experiments

4.1. Setups

The dataset used in this paper is taken from a COVID CXR Image Dataset uploaded to Kaggle [1] where the researchers created a comprehensive Chest X-ray images dataset from various research and reports. The dataset provides a posteroanterior view of chest X-ray images, with

	Custom CNN	Supervised Contrastive Learning	MiniVGG	ResNet
No Pre-process	0.845	0.875	0.798	0.912
Canny Edge Detection	0.894	0.923	0.901	0.917
Adaptive Thresholding	0.825	0.899	0.881	0.890
Shannon's Entropy	0.833	0.901	0.887	0.894

Table 1. Performance results for combinations of model and pre-process.

536 COVID-19 infected images, 668 normal images, and 619 viral pneumonia images.

Since the dataset is approximately balanced, the main evaluation criterion employed is test accuracy. A confusion matrix on the three classes is also generated for comparison and the ROC curve (one vs. all) for each class is also saved. In addition to comparing across computational models, we would also compare our model's performance with existing laboratory tests like the RT-PCR test which has around 96% accuracy [5].

Other users on Kaggle had published their models and results [3, 2] and achieved around 90% of accuracy. Since these users had published their codes, this research would only use their accuracy results as a base for comparison and instead uses two common image classification models (MiniVGG and Resnet18) as the baseline for model comparison.

4.2. Additional Experiment Details

For our custom CNN, our proposed structure parameters follow what we had from homework 5 [11]. For our proposed contrastive learning framework, we fix the temperature τ to be 0.07. We use ResNet as the backbone for our encoder network. We experiment with both dropping the projection network and not dropping the projection network at inference in Stage 2. Our finding agrees with the previous works that keeping the encoder network contributes to better performance. We did not conduct extensive hyperparameter tuning and we fix our learning rate to be 0.001, weight decay to be 0, and batch size to be 8 for Contrastive Learning Stage 1 (due to insufficient memory) and 16 otherwise.

4.3. Main Results

Examining Table. 1 we can see that our proposed custom CNN structure is the clear least accurate model in terms of prediction. This is expected because of its simple structure optimized for computing efficiency. The supervised contrastive learning structure however produced improved results in terms of accuracy compared to both our custom model and baseline models. Figure 4 shows the best epoch achieves a ROC area of 0.97. Given the relatively low epochs and training time, we can reasonably expect marginally better predictions if the model was configured to run for higher epochs.

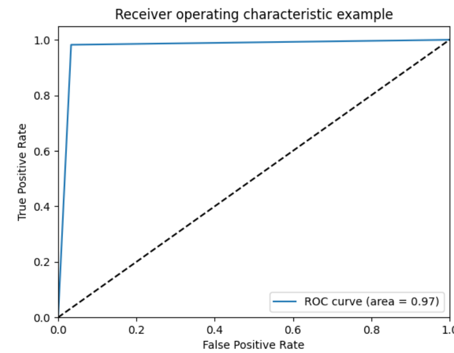

Figure 4. ROC curve for supervised contrastive learning model.

By comparing the columns of table 1, we can also compare the efficacy of each image-processing method for the purpose of priming ct-scan images for the purpose of COVID detection. The accuracies show that Canny Edge Detection has the most potential to improve accuracy on models, as it shows improvement when augmented on every model listed.

5. Conclusions

In this project, we created two machine learning architectures: custom CNN and supervised contrastive learning network to classify chest X-ray images and achieved similar accuracy results compared to previous works on the same dataset. The custom CNN has the advantages of having fewer parameters and a more efficient training time, and the supervised contrastive training network achieves the highest accuracy compared to our baseline models and comparable to the results presented by other Kaggle users.

While relatively high for purposes of machine learning, the accuracy rate is still low compared to laboratory methods. Future works can follow the model architectures mentioned in the Method section and add more image pre-processing to attempt a higher accuracy result. Through this research, we hope the two models can potentially benefit medical personnel in detecting COVID-19 faster among tested people and prevent the worsening of the illness.

References

[1] Covid cxr image dataset (research). https://www.kaggle.com/datasets/sid321axn/covid-cxr-image-dataset-research. Accessed: 2022-12-14.

[2] Covid prediction. https://www.kaggle.com/code/rich121/covid-prediction. Accessed: 2022-12-14.

[3] Detecting covid-19 from chest x-ray [by: Priyansh]. https://www.kaggle.com/code/singhalpriyansh/detecting-covid-19-from-chest-x-ray-by-priyansh. Accessed: 2022-12-14.

[4] Dhurgham Al-Karawi, Shakir Al-Zaidi, Nisreen Polus, and Sabah Jassim. Machine learning analysis of chest ct scan images as a complementary digital test of coronavirus (covid-19) patients. *MedRxiv*, 2020.

[5] Auguste Brihn, Jamie Chang, Kelsey OYong, Sharon Balter, Dawn Terashita, Zach Rubin, and Nava Yeganeh. Diagnostic performance of an antigen test with rt-pcr for the detection of sars-cov-2 in a hospital setting—los angeles county, california, june–august 2020. *Morbidity and Mortality Weekly Report*, 70(19):702, 2021.

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[7] René Cordtz, Jesper Lindhardsen, Bolette G Soussi, Jonathan Vela, Line Uhrenholt, Rasmus Westermann, Salome Kristensen, Henrik Nielsen, Christian Torp-Pedersen, and Lene Dreyer. Incidence and severeness of covid-19 hospitalization in patients with inflammatory rheumatic disease: a nationwide cohort study from denmark. *Rheumatology*, 60(SI):SI59–SI67, 2021.

[8] Nilanjan Dey, V Rajinikanth, Fuqian Shi, João Manuel RS Tavares, Luminita Moraru, K Arvind Karthik, Hong Lin, K Kamalanand, and C Emmanuel. Social-group-optimization based tumor evaluation tool for clinical brain mri of flair/diffusion-weighted modality. *Biocybernetics and Biomedical Engineering*, 39(3):843–856, 2019.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[11] Andrew Owens. Eecs 442/504 ps5: Scene recognition, 2022.

[12] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[13] Prashant Kumar Shukla, Jasminder Kaur Sandhu, Anamika Ahirwar, Deepika Ghai, Priti Maheshwary, and Piyush Kumar Shukla. Multiobjective genetic algorithm and convolutional neural network based covid-19 identification in chest x-ray images. *Mathematical Problems in Engineering*, 2021, 2021.

[14] Manu Siddhartha and Avik Santra. Covidlite: A depth-wise separable deep neural network with white balance and clahe for detection of covid-19. *arXiv preprint arXiv:2006.13873*, 2020.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.