

One Shot Model For The Prediction of COVID-19 and Lesions Segmentation In Chest CT Scans Through The Affinity Among Lesion Mask Features

Aram Ter-Sarkisov

CitAI, Artificial Intelligence Research Centre
Department of Computer Science
City, University of London

December 29, 2020

Abstract

We introduce a model that segments lesions and predicts COVID-19 from chest CT scans through the derivation of an affinity matrix between lesion masks. The novelty of the methodology is based on the computation of the affinity between the lesion masks' features extracted from the image. First, a batch of vectorized lesion masks is constructed. Then, the model learns the parameters of the affinity matrix that captures the relationship between features in each vector. Finally, the affinity is expressed as a single vector of pre-defined length. Without any complicated data manipulation, class balancing tricks, and using only a fraction of the training data, we achieve a 91.74% COVID-19 sensitivity, 85.35% common pneumonia sensitivity, 97.26% true negative rate and 91.94% F1-score. Ablation studies show that the method can quickly generalize to new datasets. All source code, models and results are publicly available on <https://github.com/AlexTS1980/COVID-Affinity-Model>.

1 Introduction

There are three main approaches to the early detection of COVID-19: reverse transcription polymerase chain reaction (RT-PCR), which is considered the golden standard for COVID-19 detection [XZZ⁺20], chest x-ray (CXR) and chest computer tomography (CT) scans. CXR is more rapid, and hence preferable at the times of high workload at radiological departments. CT scans are slower, but more accurate, because of the axial rather than frontal angle of CXR. Both methods quickly became an active area of research in Deep Learning community, with a large number of open-source datasets, such as CNCB-NCOV [ZLS⁺20] and models, such as COVIDNet [WW20], JCS [WGM⁺20], COVIDNet-CT [GWW20]. 2D lung CT analysis can also be extended to 3D, as in [HWH⁺20] extending a single slice prediction to the full scan. In addition to RT-PCR, CXR and CT scans, lungs ultrasound scans are sometimes used, as in [RMO⁺20].

Most Deep Learning algorithms predicting COVID-19 from chest CT scans use one of the three approaches to classification: general-purpose feature extractor such as ResNet or DenseNet, or a specialized one, like COVIDNet-CT mapping the input to the predicted class, [GWW20, BGCB20, LQX⁺20, YWR⁺20, SZL⁺], a combination of feature extraction and a semantic segmentation/image mask, [JWX⁺20, WGM⁺20, ZZHX20] and a combination of regional instance extraction and global (image) classification, [TS20a, TS20b]. A large number of models segment the mask of the lesions without predicting the class, a task left to the medical practitioner, e.g. [FZJ⁺20].

Each approach has certain drawbacks regardless of the achieved accuracy of the model. These drawbacks include a

small size of the dataset [BGCB20], limited scope (only two classes: COVID-19 and Common Pneumonia (CP) [SZL⁺], COVID-19 and Control [WGM⁺20], COVID-19 and non-COVID-19 [ZZHX20]), large training data requirement [GWW20], large model size [LQX⁺20, TS20a]. In [TS20a] the drawback of using a large amount of data was addressed by training a Mask R-CNN [HZRS16] model to segment areas with lesions in chest CT scans. Then, the model was augmented with a classification head that predicts the class of the image. This allowed for using a much smaller dataset for training than, e.g. [GWW20] at the cost of the size of the model.

A number of solutions uses localization of features to improve the performance and stability of the models. In [OPY20] an ensemble of majority-voting ResNet18 networks was trained on sets of randomly sampled and cropped patches from different areas of chest X-rays. In [WGM⁺20] a UNet pixel-level semantic segmentation model was used to extract image-level lesion masks to augment the classifier's feature maps. Often a combination of segmentation and classification helps with deriving saliency maps that increase the explainability of the model's results, such as in [OPY20, WGM⁺20]. For the purpose of explaining the classification results models based on Faster R-CNN [RHGS15] and Mask R-CNN [HGDG17] like COVID-CT-Mask-Net [TS20a] and Single Shot Model for COVID-19 prediction (SSM) [TS20c] demonstrate strong performance, due to Mask R-CNN's ability to detect and segment instances of regions of interest (RoIs), i.e. the ability to understand the data at an object level.

In this paper we introduce a model that learns the affinity between lesion mask features to predict COVID-19 and segment lesions. Many publications on COVID-19, e.g. [ZZX⁺20, LFBL20] analyzing the differences between COVID-19 and other types of pneumonia note that these differences, although observable, are often not statistically significant. Certain lesion features specific to COVID-19 arise simultaneously, e.g. peripheral Ground Glass Opacity (GGO): although it is observed in both COVID-19 and common pneumonia patients, it tends to be bilateral in COVID-19 patients more often. Therefore, a model that can both explicitly localize each lesions and link them in a meaningful way will be able to predict COVID-19 more accurately.

The main contribution of this study can be summarized as follows:

- High precision of lesion segmentation and prediction accuracy of COVID-19,
- Training and evaluation of the model to solve a segmentation and classification problem in one shot,
- Architecture for expressing each lesion mask as a vector
- Novel architecture of the trainable affinity matrix and affinity vector that express the affinity among lesion masks in a single vector, from which the class of the image is predicted.

To the best of our knowledge, this is the first model that uses such advanced machinery to achieve a high accuracy of COVID-19 prediction from chest CT scans.

The rest of the paper is structured as following: Section 2 introduces the segmentation and classification datasets, Section 3 the methodology of the models, in Section 4 results and analysis of experiments are presented and Section 5 concludes.

2 Data

The raw chest CT scan data is taken from CNCB-NCOV repository, [ZLS⁺20], <http://ncov-ai.big.ac.cn/download>. The raw dataset is split into two subsets: experimental data taken from COVID-19 positive patients labelled at pixel level that we use for the segmentation model, and the full data labelled at slice level that we use for the classification model. The former dataset contains a total of 3 positive pixel-level classes: Ground Glass Opacity (GGO), Consolidation (C) and normal lungs. The latter dataset contains 3 image-level classes: COVID-19, Common Pneumonia (CP) and Normal/Control.

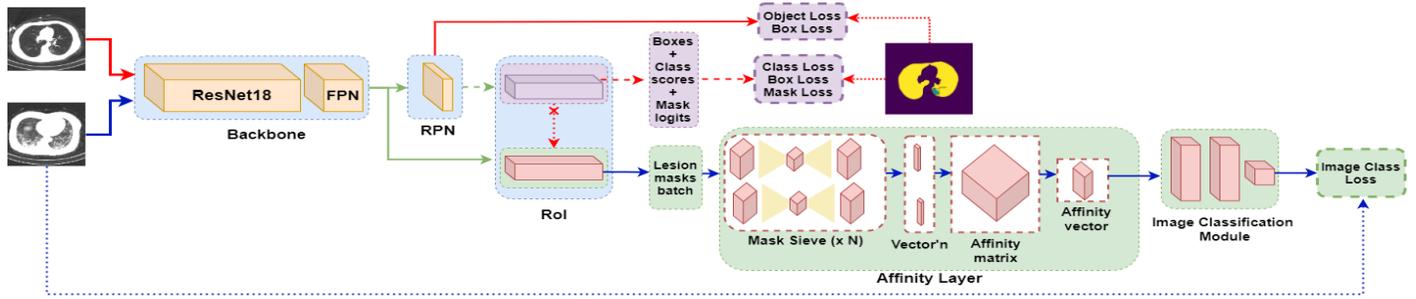


Figure 1: One Shot Affinity Model. RPN and RoI are connected to the FPN feature output only (last feature layer). Blue blocks: shared Mask R-CNN layers, purple blocks: only Mask R-CNN, green blocks: only Affinity model. Normal arrows: tensors operations, broken arrows: samples/batches, Dotted arrows: labels/loss computation. Broken+cross arrow: weight copy (segmentation to classification box and mask features). Best viewed in color.

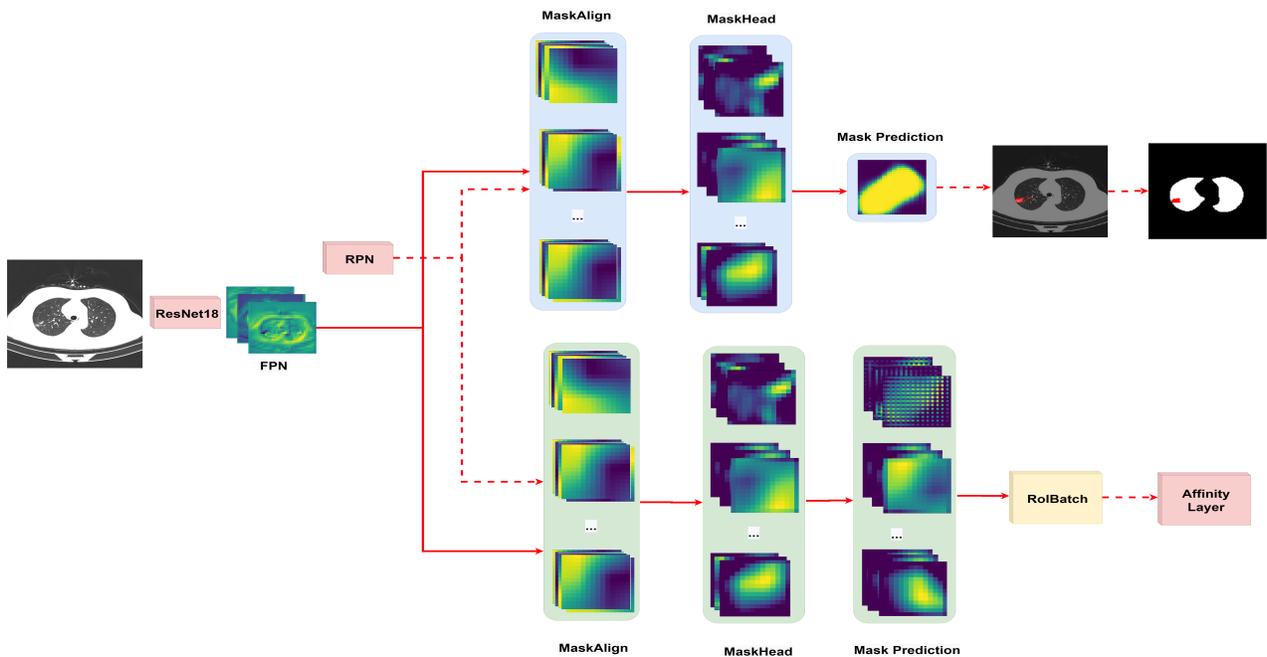


Figure 2: Segmentation (top) and Classification (bottom) lesion mask features branches. Segmentation branch outputs mask class logits for each lesion resized to the bounding box/image. Classification branch outputs a batch of lesion mask features used as an input in the Affinity layer. Normal lines: convolutional operators, broken lines: all other operators. Note all mask features in the MaskAlign and MaskHead layers are the same. Best viewed in color.

Algorithm 1: One Shot Segmentation and Classification Algorithm Using Lesions Mask Affinity

Θ_1 : set of backbone parameters,
 $\Theta_2 = \Theta_2^{\text{RPN}} \cup \Theta_2^{\text{RoI}}$: set of segmentation branch parameters
 $\Theta_3 = \Theta_3^{\text{RoI}} \cup \Theta_3^{\text{Affinity}} \cup \Theta_3^{\text{CLS}}$: set of classification branch parameters
 Λ : set of model hyperparameters
for $t= 1$ **to** T **do**
 Segmentation stage:
 Input: CT scan Image +Instances’ Masks, Boxes, Class Labels (segmentation data)
 Compute and backprop $L_{SEG}(t)$
 if $L_{SEG}(t) < L_{SEG}(t - 1)$ **then**
 | Copy weights from Θ_2^{RoI} to Θ_3^{RoI}
 end
 Update Θ_1, Θ_2
 Classification stage :
 Input: Image+Image Class Label (classification data)
 Extract lesion mask feature batch using RoI classification branch
 Vectorize lesion mask batch
 Compute affinity matrix $\mathbf{X}_2 = \mathbf{W}_2^T \mathbf{X}_1$
 Compute affinity vector $\mathbf{x}_3 = \mathbf{X}_2^T \mathbf{w}_2$
 Compute classification loss using image classification module $CLS : L_{CLS}(t) = CLS(\mathbf{x}_3)$
 Backpropagate $L_{CLS}(t)$ and update $\Theta_1, \Theta_3^{\text{Affinity}}, \Theta_3^{\text{CLS}}$
 Store $L_{Total}(t) = L_{CLS}(t) + L_{SEG}(t)$ for progress tracking
end
Final model minimizing L_{Total}

Table 1: Summary of the CNCB-NCOV [ZLS⁺20] segmentation dataset.

Split	Positive	Negative	Total
Train/Val	475	175	650
Test	71	29	100

Table 2: Summary of the CNCB-NCOV [ZLS⁺20] classification dataset and COVIDx-CT splits [GWW20].

Split	COVID-19 patients	CP patients	Normal patients	Total patients	COVID-19 slices	CP slices	Normal slices	Total slices	Ratio Train/Test
Train	300	420	144	864	12520	22061	27201	61782	2.9153
Train-ours	10	10	10	30	1000	1000	1000	3000	0.1415
Validation	95	190	47	332	4529	7400	9107	21036	-
Test	116	125	52	293	4346	7395	9450	21192	-

Summary of the segmentation dataset is presented in Table 1. We merge clean lungs masks with the background into a single negative class and GGO with C masks into a single positive class mask ('Lesions'), hence we have 1 negative and 1 positive classes. The presence of negative slices for COVID-19 positive patients is explained in the following way:

- All slices with lesions smaller than 10×10 pixels were merged with the background,
- Specifics of the manifestation of COVID-19 in chest CT scans

The train/validation/test splits for the classification model in Table 2 are taken from COVIDx-CT [GWW20]. This is done for the sake of consistency of results: negative images for positive classes were removed from the final data by the developers of COVIDx-CT.

The second row in Table 2 is the batch we sampled from the training set in COVIDx-CT. Only 3000 images in total (1000/class) were used for to train and report all models in Section 4. Validation and test splits were used in full (21036 and 21192 images resp.).

2.1 Comparison To Other Solutions

Best results in, e.g. [GWW20, OPY20, WW20] heavily depend on a number of dataset augmentation tricks. The main one is the ratio of train/test data splits (last column in Table 2). Compared to [GWW20, OPY20, BGCB20] we use a small share of the train data and evaluate the model on the whole test split. In addition to that, we did not use any of the following data balancing tricks implemented in other studies:

1. Image enhancement (histogram equalization),
2. Image manipulation (rotation, jittering, random bounding boxes, shearing, cropping, etc),
3. Class-based resampling (class balancing).

Apart from the subtraction of the global mean and division by the global standard deviation, we applied no other data manipulations to either dataset, which is one of the key strengths of our approach. All dataset interfaces used in the paper are publicly available on <https://github.com/AlexTS1980/COVID-Affinity-Model>. Additional statistical analysis of the data in Tables 2 and 1 was presented in [GWW20, TS20a, TS20b].

We selected CNCB-NCOV dataset for our investigation for the following reasons:

- Size of the classification dataset (over 104000 images),
- Three image-level classes (COVID-19 vs CP vs Normal),
- High-quality segmentation dataset of 750 images.

This is in contrast to other publicly available datasets that suffer from a number of weaknesses:

- Radiopedia (www.radiopedia.org) medical segmentation dataset contains scans from 9 patients (CNCB-NCOV:150 patients)
- UCSD dataset [ZZHX20] contains 746 images across two classes (COVID vs Non-COVID),
- COVID-CTset dataset [RAS20] contains 12058 images across two classes (COVID vs Normal),
- SARS-COV-2-Ct-Scan dataset [SAB⁺20] contains 2481 images across two classes (COVID vs CP).

3 Methodology

The overall architecture of the model is presented in Figure 1 and the training protocol in Algorithm 1.

3.1 Mask R-CNN Functionality

The model depends on Mask R-CNN [HG17] functionality to perform a number of tasks on the input images:

- Region Proposal Network (RPN): prediction of bounding box coordinates for positive regions of interest (contain objects),
- Region of Interest (RoI): extraction of RoIs of fixed size using RoIAlign pooling method, prediction of RoI's class and encoded/decoded bounding box coordinates, extraction of the lesion masks of fixed size at instance level (per bounding box),
- Extraction and ranking of a batch of positive RoIs (RoIs containing objects rather than the background) using their confidence scores.

This functionality was extensively used to train COVID-CT-Mask-Net [TS20a] and Single Shot Model (SSM) [TS20c]. These two models include several extensions of Mask R-CNN functionality for the global (image-level) prediction:

- RoI batch selection [TS20a]. The most important implication of this feature for image classification is the acceptance of all RoIs regardless of their confidence scores, which is important for Normal and CP class. A subset of positive RoIs are discarded based on the degree of overlaps and size. A fixed number of the remaining ones is ranked based on the confidence score. This batch is fed into the image classification module. This architecture converts batch of encoded boxes into a stacked feature vector.
- RoI branches [TS20c]. Augmentation of RoI to include the third parallel branch (in addition to detection and segmentation branches in Mask R-CNN): classification branch with the architecture identical to the detection branch. Detection branch is trained normally, and its weights are copied into the classification branch if the detection+segmentation error improves. Classification branch is not updated in any other way. This allows the classification branch to understand the input at an instance level (boxes+box scores) and output encoded box coordinates and their class scores, which are essential to the image classification.

Next, we show how we extend this methodology to the extraction of lesion masks and the derivation of affinity between their features.

3.2 Lesion Mask Features Branch in RoI

We augment the RoI architecture with the fourth parallel branch that has the same architecture as Mask R-CNN mask branch, see Figure 2. Mechanism of the adaptation of this layer is the same as of the classification branch [TS20c]: the weights are copied from the mask branch if the segmentation mask loss improves. No other loss is computed, see Equation 8. The only difference from Mask R-CNN is that instead of mask class logits this branch outputs a set of lesion mask features of fixed size, $C \times H \times W$ for each positive RoI: C is the number of features in each lesion mask, $H \times W$ is the height and width of each map.

We apply the RoI batch selection trick described above to obtain a fixed-size batch of ranked lesion mask features, size $B \times C \times H \times W$, where B is the size of the batch.

From now on we apply the term 'Lesion mask' to every positive RoI regardless of the image class and confidence score. Obviously Negative images do not contain any lesions, but 'lesions' with a negligible (yet positive) confidence score.

3.3 Affinity Layer

The mask lesions batch size $B \times C \times H \times W$ is the input in the Affinity layer, which is at the core of our methodology. Its job is to vectorize this batch, derive affinity among all lesion masks and express it in a single vector of fixed length that is used to predict the image class. Affinity for the purpose of this problem is defined as the relationship among the lesion masks' features.

Denoting each lesion \mathbf{x}_k , we derive the affinity only between the same features, i.e. between feature j in each lesion $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_B$. The motivation of this approach comes from the fact that at this stage features have high semantic values, but, due to the ranking, lesion masks can have very different confidence scores and therefore extracted from very different areas in scans, varying from truly affected by GGO and C, to healthy lungs and background. For example, affinity at j^{th} feature between the first and the second-ranked lesion mask in a Negative image will be very different from the affinity between these lesions in COVID-19 positive image. This is the justification for the shape of the affinity matrix (see below), as multiple affinities (at most C') can exist between features.

Affinity layer works in four stages:

- Mask sieve,
- Mask Vectorization,
- Affinity matrix,
- Affinity vector

In the mask sieve step, batches are processed independently, as in RoI, and their size is first downsampled from $H \times W$ to $H/2 \times W/2$, then upsampled back to $H \times W$ a total of N times. The difference from the adaptation mechanism in RoI is that the weights of the sieve are trained wrt image loss, see Equations 9 and 10. The output size of the sieve is the same as the input size. In the next step the features are vectorized, i.e. their dimensions are downsampled from $H \times W$ to 1×1 maintaining the batch and channel dimensions, so the input in the affinity matrix is size $B \times C$.

3.4 Affinity Matrix

This is the key step in the model. We refer to this method as semi-supervised, because at this stage vectorized lesion masks are unlabelled. Labeling is only effected at the image level, and the supervised loss computation is only done at the image level, and extended through backpropagation to the affinity layer.

Each vectorized lesion mask \mathbf{x}_k consists of C features, so, by composing them row-wise, the lesion mask feature matrix \mathbf{X}_1 can be written as an array size $B \times C$ of row vectors \mathbf{x}_k , Equation 1. In order to extract affinities among the lesions masks, we introduce a trainable *affinity matrix* \mathbf{W}_1 , which can learn C' affinities for each feature.

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_B \end{bmatrix} \quad (1)$$

$$\mathbf{W}_1 = \left[\mathbf{w}_{1,1} : \mathbf{w}_{1,2} : \dots : \mathbf{w}_{1,C'} \right] \quad (2)$$

$$\mathbf{X}_2 = \mathbf{W}_1^T \otimes \mathbf{X}_1 = \left[\bar{\mathbf{x}}_{2,1} : \bar{\mathbf{x}}_{2,2} : \dots : \bar{\mathbf{x}}_{2,C'} \right] \quad (3)$$

In Equation 2 the trainable affinity matrix \mathbf{W}_1 has dimensionality $B \times C'$ and can be written as a composition of column vectors $\mathbf{w}_{1,r}$, each length B . The job of each vector $\mathbf{w}_{1,r}$ in Equation 2 is to learn the r^{th} affinity for the j^{th} feature across

all lesion masks by taking the vector product with the feature, Equations 3. $x_{2,r,j}$ in Equation 4 is the strength of r^{th} affinity in j^{th} feature.

$$x_{2,r,j} = \mathbf{w}_{1,r} \otimes \mathbf{x}_{1,j} \quad (4)$$

Matrix \mathbf{X}_2 can be used for the purpose of analyzing the progress of the training protocol, see Section 4.3.

3.5 Affinity Vector

Once \mathbf{X}_2 is obtained, we can take its matrix-vector product with the *affinity vector* \mathbf{w}_2 that scales the affinities and maps the features back into the feature space, Equation 5.

$$\mathbf{x}_3 = \mathbf{X}_2^T \otimes \mathbf{w}_2 = \begin{bmatrix} \bar{x}_{2,1}\mathbf{w}_2 \\ \bar{x}_{2,2}\mathbf{w}_2 \\ \vdots \\ \bar{x}_{2,C}\mathbf{w}_2 \end{bmatrix} = \begin{bmatrix} \bar{x}_{3,1} \\ \bar{x}_{3,2} \\ \vdots \\ \bar{x}_{3,C} \end{bmatrix} \quad (5)$$

\mathbf{x}_3 expresses the affinity among B vectorized lesion mask features in a single vector, considering the affinity for each feature independently of other features.

3.6 Image Classification Module and Loss Function

Vector \mathbf{x}_3 is the input in the final image classification module with two fully connected layers and the output class logits layer with 3 neurons (one/class).

The model solves a segmentation and classification problem simultaneously using a linear combination of two loss functions, Equation 6.

$$L_{Total} = L_{SEG} + L_{CLS} \quad (6)$$

$$L_{CLS} = -t_k^{Cl} \log \sigma(h_k(\mathbf{x})) - \sum_{k \neq Cl} (1 - t_k^{Cl}) \log(1 - \sigma(h_k(\mathbf{x}))) \quad (7)$$

$$t_k^{Cl} = \begin{cases} 1 & \text{if } k=Cl \\ 0 & \text{else} \end{cases}$$

In Equation 6 L_{SEG} is the same as in Mask R-CNN [HGDG17]. Equation 7 is per-class binary cross-entropy. This means that we compute 3 image class loss values. Here σ is sigmoid function, $h_k(\mathbf{x})$ is a logit output of the class neuron for some linear input map \mathbf{x} , Cl is the correct class of the input image. This loss function is straightforward for such an advanced model, which is also one of its advantages.

The difference between the mask features at RoI stage and Mask Sieve stage can be expressed using the partial derivatives wrt to L_{CLS} and L_{SEG} , Equations 8-10,

$$\frac{\partial L_{CLS}}{\partial \mathbf{W}_{Masks}^{RoI}} = \frac{\partial L_{SEG}}{\partial \mathbf{W}_{Masks}^{RoI}} = 0 \quad (8)$$

$$\frac{\partial L_{SEG}}{\partial \mathbf{W}_{Masks}^{MaskSieve}} = 0 \quad (9)$$

$$\frac{\partial L_{CLS}}{\partial \mathbf{W}_{Masks}^{MaskSieve}} \neq 0 \quad (10)$$

where \mathbf{W} is the matrix of weights in the corresponding layer (RoI or Mask Sieve).

Table 3: Model sizes (million weights).

# Affinities	Affinity Layer	Backbone +FPN	RPN+RoI (Classification branch)	Total
32	4.57M			14.18M
64	4.57M			14.18M
128	4.58M	3.44M	6.17M (3.06M)	14.19M
256	4.58M			14.19M
512	4.59M			14.20M

4 Experimental results

4.1 Implementation details

We trained the model using Adam optimizer with the fixed learning rate of $1e - 5$ and regularization parameter $1e - 3$. Lesion mask features batch size was set to 8 to minimize computation time (also, increasing it did not lead to a comparable payoff in the accuracy). Hence, each lesion mask feature batch is $8 \times 256 \times 28 \times 28$, so the vectorized input in the Affinity matrix is 8×256 . The number of tested affinities was 32, 64, 128, 256, 512, the resulting model sizes are presented in Table 3. The number of stages of the mask sieve was set to 3 (further increase did not result in a noticeable improvement). Batch size was set to 1 for both problems. Details of the training procedure are presented in Algorithm 1.

Each model was trained for 100 epochs, which took about 20 hours. All experiments were run on a GPU with 8Gb of VRAM.

4.2 Segmentation Results

For the explanation of the accuracy metrics and comparison, see [LMB⁺14], as we use MS COCO’s average precision (AP) at two Intersect over Union (IoU) threshold values (0.5 and 0.75) and mean AP across 10 IoU thresholds between 0.5 and 0.95 with a 0.05 step (main MS COCO metric). The overlap is computed at the mask rather than bounding box level (segmentation problem). All mask logits predicted by the model are rescaled to the predicted bounding box, embedded in the image size $h \times w$ and filtered through a simple 0–threshold (i.e. all logits > 0 are positive mask predictions). Each predicted and gt mask (m_1, m_2) are vectorized before IoU computation, Equation 11.

$$\begin{aligned}
 I(m_1, m_2) &= \sum_{k=1}^{h \times w} p_k(m_1) p_k(m_2) \\
 S(m_j) &= \sum_{k=1}^{h \times w} p_k(m_j) \\
 p_k &= \begin{cases} 1 & \text{object} \\ 0 & \text{background} \end{cases} \\
 IoU(m_1, m_2) &= \frac{I(m_1, m_2)}{S(m_1) + S(m_2) - I(m_1, m_2)} \tag{11}
 \end{aligned}$$

To account for the negative images, we set the precision to 0 if there is at least a single detection, and to 1 if there are no detections. Results are presented in Table 4. At present the highest mAP on MS COCO 2017 segmentation leaderboard is 51.3% (AP at the range of IoU thresholds between 0.5 and 0.95), and the mean about 39.98%, so our results are at par with MS COCO. The model with $C' = 64$ affinities achieved a 0.42 mAP, which is well above the MS COCO average.

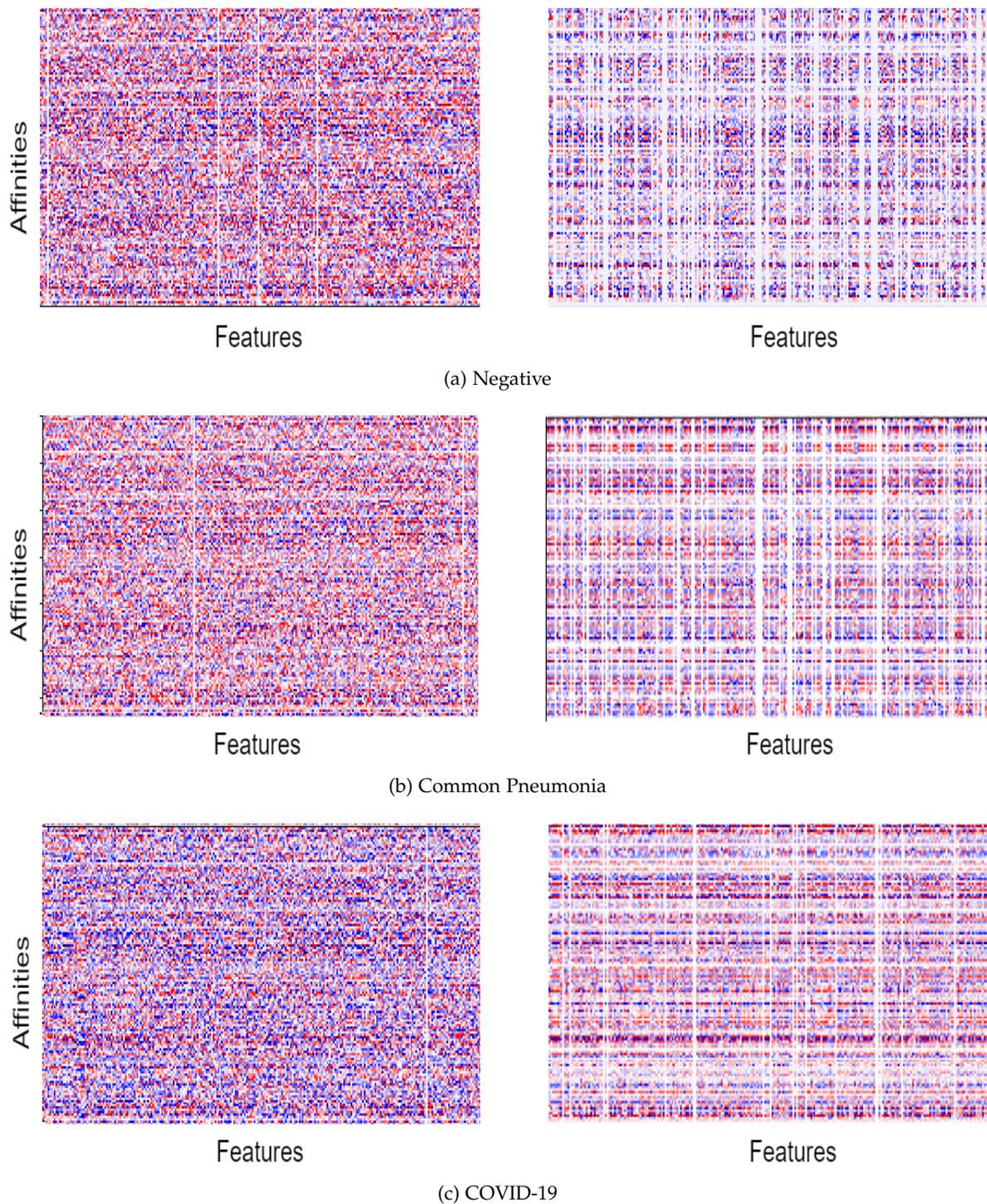


Figure 3: Derived affinities among features, X_2 matrix in Equation 3. Left column: after 1 epoch, right column: after 100 epochs. Color gamma is normalized between 255 (red, strong positive affinity) and -255 (blue, strong negative affinity). Best viewed in color.

Table 4: Average Precision on the segmentation data test split (100 images). Best results in bold.

# Affinities	AP@0.5 IoU	AP@0.75 IoU	AP@[0.5:0.95]IoU
32	0.614	0.382	0.395
64	0.603	0.414	0.422
128	0.569	0.350	0.385
256	0.560	0.347	0.386
512	0.548	0.343	0.386

Table 5: Accuracy results on the COVIDx-CT test split (21191 images). Best results in bold.

# Affinities	COVID-19	Common Pneumonia	Negative	F ₁ score
32	89.39%	80.25%	98.96%	90.30%
64	90.68%	83.60%	97.15%	91.00%
128	86.91%	95.65%	95.45%	93.80%
256	91.74%	85.35%	97.26%	91.94%
512	90.27%	84.53%	99.41%	92.34%

4.3 Classification Results

To compute classification accuracy, we used per-class c sensitivity, specificity and F₁score, Equations 12-14.

$$Sens(c) = \frac{TP}{TP + FN} \quad (12)$$

$$Spec(c) = \frac{TN}{TN + FP} \quad (13)$$

$$F_1score = 2 \sum_{c=1}^C w_c \times \frac{Sens(c) \times Spec(c)}{Sens(c) + Spec(c)} \quad (14)$$

where $w(c)$ is the share of each class in the test split.

Results for each number of affinities C' are presented in Table 5. Without any complicated data manipulations, the model with 256 affinities achieved a 91.74% COVID-19 sensitivity, and F₁score of 91.94%, and the model with 128 affinities an F₁score of 93.80%, common pneumonia sensitivity of 95.65% and COVID-19 sensitivity of 86.91% which is alongside the highest reported results in the COVID-19 Deep Learning studies for datasets of this size.

Matrix X_2 for the model with $C' = 128$ and 3 different input images is visualized in Figures 3a-3c after training for 1 and 100 epochs (in the latter case the input image is always correctly classified). Each element (i, j) of the matrix is the strength of the i^{th} affinity for j^{th} feature element across all lesion vectors. The structure of the matrix in Figure 3a has a greater presence of inactive ('pale') affinities, but several affinities are consistently negative across all features. Figures 3b and 3c are similar in the way that they have both a large number of strong and weak affinities, a large number of affinities is consistently strong across all features, and also a large number of affinities change sign and magnitude across vectors.

4.4 Ablation studies

One of the challenges of fast and reliable COVID-19 diagnosis is the generalization of models to other datasets. We extend our findings to iCTCF-CT open-source dataset [NLY⁺20], <http://ictcf.biocuckoo.cn> with Negative and COVID-19 classes, summarized in Table 6. We split the data randomly into 600 training/validation and 12976 test images (note the class imbalance in the test split) achieving the test/train+val ratio of 21.62. We finetuned each models to the data for 20

Table 6: Summary of the iCTCF-CT [NLY+20] classification dataset.

Split	COVID-19	Negative	Total
Train/Val	300	300	600
Test	3701	9275	12976

Table 7: Accuracy results on the iCTCF-CT test split (12976 images). Best results in bold.

# Affinities	COVID-19	Negative	F ₁ score
32	92.11%	80.31%	83.67%
64	86.73%	94.20%	92.06%
128	88.88%	83.85%	85.27%
256	77.41%	93.33%	88.78%
512	90.49%	89.96%	90.11%

epochs, which took about 25min on a GPU. No changes were applied to the models' architecture.

Ablation study results in Table 7 confirm that the affinity model approach generalizes quickly and achieves a high accuracy on the new data.

5 Conclusions

In this paper we presented a novel methodology for the computation of affinity among lesion mask features for the simultaneous segmentation of lesions and prediction of COVID-19 from chest CT scans. At the core of the approach is a trainable Affinity matrix that captures different relationships between features from the lesion masks in a semi-supervised manner. Using a small fraction of the CNCB-NCOV training data, we achieved several strong results on the test split with 21192 images, including the model with 256 affinities that achieved a 91.74% COVID-19 sensitivity and 91.71% F1-score, which is higher than in the majority of other studies. Additional ablation studies on iCTCF demonstrates the ability of our approach to quickly generalize to the new data.

Deep investigation of the structure and the effect of the affinity matrix will be the focus of our future research. All source code, models and results are publicly available on <https://github.com/AlexTS1980/COVID-Affinity-Model>

References

- [BGCB20] Charmaine Butt, Jagpal Gill, David Chun, and Benson A Babu. Deep learning system to screen coronavirus disease 2019 pneumonia. *Applied Intelligence*, pages 1–7, 2020.
- [FZJ⁺20] Deng-Ping Fan, Tao Zhou, Ge-Peng Ji, Yi Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Medical Imaging*, 2020.
- [GWW20] Hayden Gunraj, Linda Wang, and Alexander Wong. Covidnet-ct: A tailored deep convolutional neural network design for detection of covid-19 cases from chest ct images. *arXiv preprint arXiv:2009.05383*, 2020.
- [HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

- [HWH⁺20] Zhongyi Han, Benzhenq Wei, Yanfei Hong, Tianyang Li, Jinyu Cong, Xue Zhu, Haifeng Wei, and Wei Zhang. Accurate screening of covid-19 using attention based deep 3d multiple instance learning. *IEEE Transactions on Medical Imaging*, 2020.
- [HZRS16] 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.
- [JWX⁺20] Shuo Jin, Bo Wang, Haibo Xu, Chuan Luo, Lai Wei, Wei Zhao, Xuexue Hou, Wenshuo Ma, Zhengqing Xu, Zhuozhao Zheng, et al. Ai-assisted ct imaging analysis for covid-19 screening: Building and deploying a medical ai system in four weeks. *medRxiv*, 2020.
- [LFBL20] Xiao Li, Xu Fang, Yun Bian, and Jianping Lu. Comparison of chest ct findings between covid-19 pneumonia and other types of viral pneumonia: a two-center retrospective study. *European radiology*, pages 1–9, 2020.
- [LMB⁺14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [LQX⁺20] Lin Li, Lixin Qin, Zeguo Xu, Youbing Yin, Xin Wang, Bin Kong, Junjie Bai, Yi Lu, Zhenghan Fang, Qi Song, et al. Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct. *Radiology*, 2020.
- [NLY⁺20] Wanshan Ning, Shijun Lei, Jingjing Yang, Yukun Cao, Peiran Jiang, Qianqian Yang, Jiao Zhang, Xiaobei Wang, Fenghua Chen, Zhi Geng, et al. Open resource of clinical data from patients with pneumonia for the prediction of covid-19 outcomes via deep learning. *Nature biomedical engineering*, pages 1–11, 2020.
- [OPY20] Yujin Oh, Sangjoon Park, and Jong Chul Ye. Deep learning covid-19 features on cxr using limited training data sets. *IEEE Transactions on Medical Imaging*, 2020.
- [RAS20] Mohammad Rahimzadeh, Abolfazl Attar, and Seyed Mohammad Sakhaei. A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. *medRxiv*, 2020.
- [RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [RMO⁺20] Subhankar Roy, Willi Menapace, Sebastiaan Oei, Ben Luijten, Enrico Fini, Cristiano Saltori, Iris Huijben, Nishith Chennakeshava, Federico Mento, Alessandro Sentelli, et al. Deep learning for classification and localization of covid-19 markers in point-of-care lung ultrasound. *IEEE Transactions on Medical Imaging*, 2020.
- [SAB⁺20] Eduardo Soares, Plamen Angelov, Sarah Biaso, Michele Higa Froes, and Daniel Kanda Abe. Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *medRxiv*, 2020.
- [SZL⁺] Ying Song, Shuangjia Zheng, Liang Li, Xiang Zhang, Xiaodong Zhang, Ziwang Huang, Jianwen Chen, Huiying Zhao, Yusheng Jie, Ruixuan Wang, Yutian Chong, Jun Shen, Yunfei Zha, and Yuedong Yang. Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images. *medRxiv*.
- [TS20a] Aram Ter-Sarkisov. COVID-CT-Mask-Net: Prediction of COVID-19 from CT Scans Using Regional Features. *medRxiv*, 2020.
- [TS20b] Aram Ter-Sarkisov. Detection and segmentation of lesion areas in chest CT scans for the prediction of COVID-19. *medRxiv*, 2020.
- [TS20c] Aram Ter-Sarkisov. Single-shot lightweight model for the detection of lesions and the prediction of covid-19 from chest ct scans. *medRxiv*, 2020.

- [WGM⁺20] Yu-Huan Wu, Shang-Hua Gao, Jie Mei, Jun Xu, Deng-Ping Fan, Chao-Wei Zhao, and Ming-Ming Cheng. Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation. *arXiv preprint arXiv:2004.07054*, 2020.
- [WW20] Linda Wang and Alexander Wong. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *arXiv preprint arXiv:2003.09871*, 2020.
- [XZZ⁺20] Xingzhi Xie, Zheng Zhong, Wei Zhao, Chao Zheng, Fei Wang, and Jun Liu. Chest ct for typical 2019-ncov pneumonia: relationship to negative rt-pcr testing. *Radiology*, page 200343, 2020.
- [YWR⁺20] Tao Yan, Pak Kin Wong, Hao Ren, Huaqiao Wang, Jiangtao Wang, and Yang Li. Automatic distinction between covid-19 and common pneumonia using multi-scale convolutional neural network on chest ct scans. *Chaos, Solitons & Fractals*, 140:110153, 2020.
- [ZLS⁺20] Kang Zhang, Xiaohong Liu, Jun Shen, Zhihuan Li, Ye Sang, Xingwang Wu, Yunfei Zha, Wenhua Liang, Chengdi Wang, Ke Wang, et al. Clinically applicable ai system for accurate diagnosis, quantitative measurements, and prognosis of covid-19 pneumonia using computed tomography. *Cell*, 2020.
- [ZZHX20] Jinyu Zhao, Yichen Zhang, Xuehai He, and Pengtao Xie. Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*, 2020.
- [ZZX⁺20] Wei Zhao, Zheng Zhong, Xingzhi Xie, Qizhi Yu, and Jun Liu. Ct scans of patients with 2019 novel coronavirus (covid-19) pneumonia. *Theranostics*, 10(10):4606, 2020.