One Shot Model For COVID-19 Classification and Lesions Segmentation In Chest CT Scans Using LSTM With Attention Mechanism

Aram Ter-Sarkisov
CitAI Research Center,
City, University of London
e-mail: alex.ter-sarkisov@city.ac.uk

***Abstract*—We present a model that fuses lesion segmentation with Attention Mechanism to predict COVID-19 from chest CT scans. The model segments lesions, extracts Regions of Interest from scans and applies Attention to them to determine the most relevant ones for image classification. Additionally, we augment the model with Long-Short Term Memory Network layers that learn features from a sequence of Regions of Interest before computing attention. The model is trained in one shot for both problems, using two different sets of data. We achieve 0.4683 mean average precision for lesion segmentation, 95.74% COVID-19 sensitivity and 98.15% class-adjusted F1 score for image classification on a large CNCB-NCOV dataset. Source code is available on https://github.com/AlexTS1980/COVID-LSTM-Attention.**

I. Introduction

Coronavirus (COVID-19) is an ongoing global pandemic that has taken so far over 5.3M lives worldwide as of December 2021 with the crisis worsening in some countries, measured both by the number of deceased and the number of new cases (https://www.worldometers.info/coronavirus). The pandemic caused a complete or partial lockdown in most countries across the planet and led to a previously unseen pressure on healthcare, with the radiology departments workload exceeding their capacity and manpower.

Analysis of chest CT scans using Deep Learning (DL) can provide assistance to the radiology personnel in many ways. One of them is the reduction of time it takes to process a scan slice from roughly 20 minutes to a few seconds and less [1]. DL algorithms can both rule out clear true positives, and draw the personnel's attention to suspicious images, e.g. by detecting and segmenting lesions. This may result in two types of errors that the algorithm can possibly make: failure to identify suspicious areas in scans (false negative) or a false alarm (false positive) due to the misclassification of images with clean lungs as COVID-19. One of the specific challenges that the personnel, and, therefore, DL algorithms, face is the misclassification of COVID-19 into other types of pneumonia, due to a large number of overlaps between the ways these diseases manifest in chest CT scans.

Existing Deep Learning methodology analyzing chest CT scans has two main limitations: either it relies on large amounts of data (and data manipulation tricks) to train the model or the model was both trained and evaluated on small amounts of data, hence the solution's ability to extend to larger datasets is questionable. Another problem that, to the best of our knowledge, all DL solutions suffer from, is transferability of results to other datasets without additional finetuning/transfer learning, something that models like Faster R-CNN or Mask R-CNN do not have a problem with due to the training on general-purpose datasets like MS COCO 2017 and Pascal VOC 2012.

One of the approaches in the analysis of images is the extraction of Regions of Interest (RoIs) containing class- and object-specific information expressed in mask features. This can be done through either semantic [2] or instance [3] segmentation of objects. In COVID-19 literature, there is a large number of models that combine semantic segmentation of lesions and CT classification, e.g. [1], [4].

The novelty and contribution of our work can be summarized in the following way:

1) Advanced architecture with an Attention Layer that learns class-relevant RoIs. Additionally, this architecture is augmented with LSTM layer that uses a batch of RoIs ranked by the Euclidean distance from the origin,
2) RoIs are expressed by their mask feature maps instead of box coordinates. Mask feature maps have a richer expression, and they consist of a large number of features, and contain more accurate information about lesions than box coordinates and confidence scores,
3) We run a large number of experiments and ablation studies for Attention-only and LSTM with attention architectures

and compare them to a large suite of benchmark models. Our best model achieves 0.4683 mean average precision on lesion segmentation problem, 95.74% COVID-19 sensitivity and 98.15% F1 score in image classification, which are among the best results on a dataset of this size

The rest of the paper is structured in the following way: Section II discusses the related literature, Section III introduces the data and details of the attention-based methodology, Section IV discusses experimental setup, results and comparison to a suite of baseline models, and also ablation studies. Limitation of the COVID-19 methodology and our approach are discussed in Section V. Section VI concludes.

II. Related Work

Mask R-CNN [3] is the state-of-the-art instance segmentation model based on the object detector Faster R-CNN [5]. It predicts the objects' bounding boxes, classes and masks independently, thus improving accuracy compared to semantic segmentation model like Fully Convolutional Net (FCN, [2]). The key steps of Mask R-CNN are backbone model that extracts image-level features, Region Proposal Net (RPN) that predicts bounding boxes and objects and Region of Interest (RoI) layer that refines bounding boxes, predicts classes and object masks, see [3] for the details. Backbone model consists of two stages: backbone feature extractor, e.g. ResNet50, [6] and Feature Pyramid Net, FPN, [7].

Long short-term memory network (LSTM, [8]) is one of the most popular recurrent neural networks (RNNs) used to analyze and extract features from sequential data. In terms of application to COVID-19 diagnosis, in [9] a combined convolutional neural net and LSTM was presented, in which LSTM takes the last features output of ConvNet (dimensions $512 \times 7 \times 7$) as an input, and LSTM's final fully connected layer predicts the class of the image (COVID-19, Common Pneumonia and Control).

Attention mechanism is one of the most active research topics in deep learning at present. It was first introduced in [10] in the form of global (connection to all encoder states) and local (connection to a window of outpus). Its functionality is based on the encoder-decoder architecture for a wide range of sequence-based problems, and the mechanism is used to weigh the effect and the relationship of the encoder's output features. Typically, weighing is done by computing softmax distribution over the outputs of the encoder to determine the most and least relevant features or outputs.

There is a number of well-received publications that use a form of attention for COVID-19 prediction and lesion segmentation. In [11] a model with residual connections and attention-aware units was used to predict COVID-19 vs Negative. In [12] attention is computed between convolution maps from two different branches of the model: 2- and 3-class problem classification branches.

The most relevant to our study are [13] that trains Mask R-CNN for lesions segmentation and a classifier for image classification and [14] that improves this architecture by fusing segmentation and classification functionality in one model. RoI layer has therefore two branches: segmentation branch (box+class and mask), and classification branch. Its architecture is identical to the segmentation branch, and it also shares weights with it. During classification training and evaluation, it detects lesions in input images that are used to classifty the whole image.

A. LSTM for object detection and image classification

Recently, a number of studies fused LSTM and ConvNets for image segmentation and object detection problems. In [15], an RNN was applied to a ConvNet's feature maps to classify whole images. In [16] box coordinate and class of the object prediction are done through fusion of Faster R-CNN and LSTM. The order of the object's parts input into LSTM is random. As the authors point out, other ordering rules had little effect on the model's accuracy. In [17] to detect masks of doctored areas in images, input image is split into a number of non-overlapping boxes. Input in LSTM uses Hilbert curve, which sets up the order of square areas, so the order of inputs is determined by the location.

III. Methodology

For both problems, like in [13], [14] we use CNCB-NCOV dataset with 3 classes: COVID-19, Common Pneumonia (CP) and Normal, and COVIDx-CT splits from [18], except that our training data has only 3000 observations (1000/class) instead of 61037 like in [18]. Test split with 21191 images was used in full. For the segmentation problem, 650 images from CNCB-NCOV dataset were used for training and validation, and 100 for testing. At lesion level, there are 3 classes: clean lungs, Ground Glass Opacity and Consolidation, so the segmentation branch learns to predict their masks.

A. Overall model

The core idea of the study is to investigate the sequence of Regions of Interest ranked by the Euclidean distance from the origin, i.e. their location rather than confidence scores as in [13]; as mentioned before, similar approaches were implemented in [15] - [17]. Our approach was also motivated by COVID-19 studies that established a range of similarities between COVID-19 and Common Pneumonia (CP) and subtle difference between them, e.g. [19].

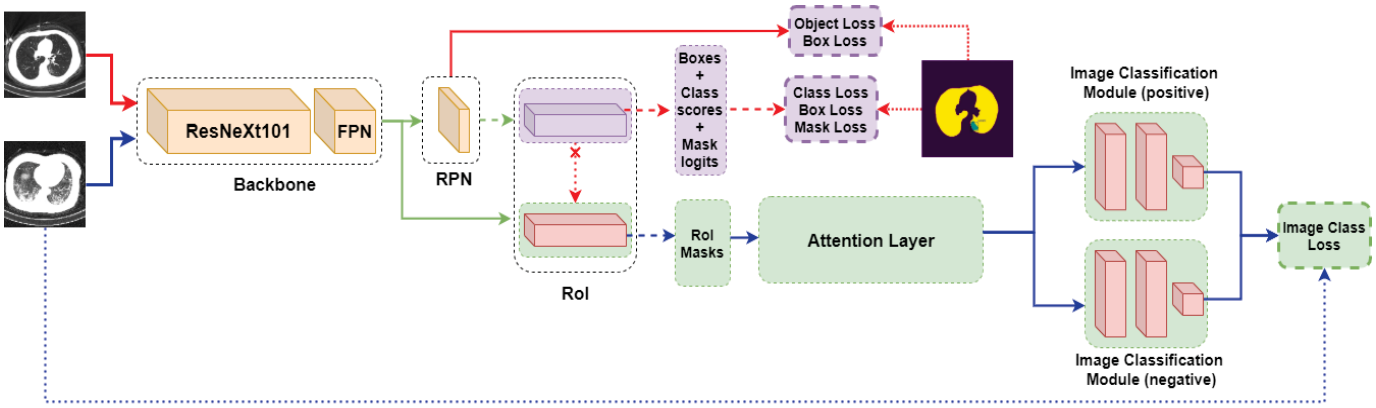


Fig. 1: One Shot Model with the Attention layer, Figure 2. Normal arrows: data, Broken arrows: batches or samples, dotted arrows: labels. Purple layers: segmentation, green layers: classification, beige layers: shared between these two. Broken layer boundaries: loss computation. Two classification layers are used in the architecture with two LSTM layers in the Attention layer. Other architectures have a single classifier. Best viewed in color.

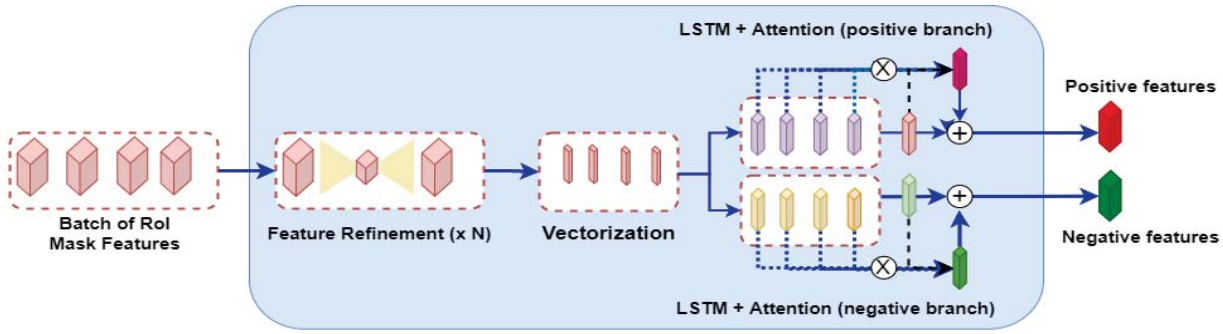


Fig. 2: Attention layer with two parallel LSTM+Attention branches (one for class-relevant, one for class-irrelevant RoIs. In the model with a single LSTM, class-irrelevant LSTM+Attention layer is deleted, and it outputs only one feature vector. Model without LSTM only uses Attention for RoIs (vectorization layer output) and outputs a single feature vector. Best viewed in color.

In this study we introduce the Attention mechanism that learns the importance of RoIs for image classification. Often, for sequential problems, a combination of LSTM and Attention mechanism is used, in which the output of LSTM at each step is weighed by softmax probability. Therefore, in this paper we investigate three architectures: base Attention model, single LSTM layer+Attention and two LSTM layers+Attention. The overall architecture of the model is shown in Figure 1.

The architecture of all models consists of the following layers:

1) Backbone feature extractor + Feature Pyramid Net (FPN),
2) Region Proposal Network layer (RPN, [3]),
3) Region of Interest layer (RoI, [3], [14]) with two branches, segmentation and classification branches,
4) Attention Layer that varies depending on the chosen Attention architecture,
5) Image classification module that also depends on the Attention architecture.

The first three layers are identical across all Attention models. RoI layer architecture is same as the one used in [14]: two parallel branches, one for segmentation problem, and one for classification. Functionality of the segmentation branch is discussed in details in [14].

In this study classification branch has two important properties:

1) For the image classification, RoI layer outputs a batch of mask features with dimensionality $\beta \times C \times H \times W$: β is the batch size, C is the number of channels (feature maps), H, W are height and width of each feature map, see [14]. For simplicity, we refer to RoI mask features as RoIs.

2) The solution here ranks RoIs using Euclidean distance from the origin to the RoI's bounding box instead of confidence scores. Note that this approach uses absolute distance from the origin to the object only to assign the rank to the RoI mask features of the object. Distance value itself is not used as an input or in any other way in the model.

We use the hack introduced in [14] for the classification branch: instead of training its weights, they are copied from

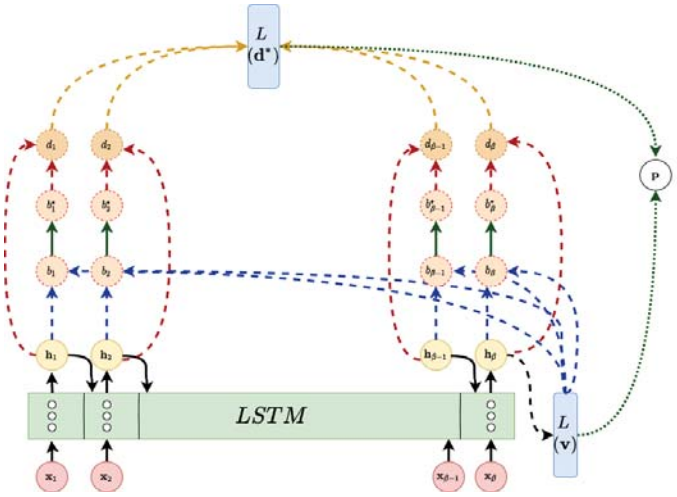


Fig. 3: LSTM with Attention. Normal black arrows: LSTM inputs and recurrent connections, broken arrows: matrix-vector product (black: $\mathbf{v}$, Equation 11, blue: $\mathbf{b}$, Equation 12, red: $\mathbf{d}$, Equation 15, yellow: $\mathbf{d}^*$, Equation 16), normal green arrows:softmax, Equations 13 and 14, dotted green arrows: elementwise summation, $\mathbf{p}$, Equation 17. Normal circles: vectors, broken circles: scalars. L:fully connected layers. Best viewed in color.

the segmentation branch, hence classification branch has the same functionality, albeit it is used for image classification.

B. RoI feature refinement and vectorization

In the context of COVID-19 prediction, this layer was first introduced in [14]. In this stage we improve the expression and strength of useful features in RoIs. The main difference from the classification branch in RoI layer, is that the weights in this stage are trainable using image-level loss. RoIs are downsized, upsized, downsampled and upsampled a total of N times. The final output has the same dimensionality as the input, but with features more relevant to the image classification rather than segmentation problem. Next, we reduce the dimensionality of each RoI from a feature map to a vector: first, from $C \times H \times W$ to $C \times \frac{H}{2} \times \frac{W}{2}$, and then to $C \times 1 \times 1$, i.e. the batch dimensionality becomes $\beta \times C \times 1 \times 1$.

C. RoI Attention Model

In Equation 1 each $\mathbf{x_k}$ is the RoI row vector and $\mathbf{X}$ is matrix with dimensions $\beta \times C$, and attention is computed for each RoI.

$$\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2} \dots \mathbf{x}_\beta] \rightarrow \mathbf{x}^* \tag{1}$$

$$\mathbf{z} = Linear(\mathbf{x}^*) \tag{2}$$

$$\mathbf{a} = \mathbf{X}\mathbf{z} \tag{3}$$

$$a_k^* = \frac{e^{a_k}}{\sum_{k=1}^{\beta} e^{a_k}} \tag{4}$$

$$\mathbf{a}^* = [a_1^*, a_2^*, \dots a_\beta^*] \tag{5}$$

$$\mathbf{c} = \mathbf{X^T}\mathbf{a}^* \tag{6}$$

$$\mathbf{c}^* = Linear(\mathbf{c}) \tag{7}$$

$$\mathbf{o} = \mathbf{c}^* \oplus \mathbf{z} \tag{8}$$

In the first step, $\mathbf{X}$ is reshaped to a vector $\mathbf{x}^*$ with dimensionality $\beta \times C$, i.e., we transform a batch into a single vector. We need to do this reshaping in order to obtain a single Attention vector later on. A fully connected trainable layer takes it as an input and outputs a vector of features $\mathbf{z}$ with C dimensions, Equation 2.

Again, we have to keep this dimensionality because of Attention computation in Equation 3: we take a matrix-vector product of $\mathbf{X}$ and $\mathbf{z}$ to obtain a vector of weights $\mathbf{a}$ (Equation 3) that is next scaled using softmax distribution $\mathbf{a}^*$ (Equation 4). In Equation 3 matrix-vector product is taken for each RoI, so $\mathbf{a}$ has dimensionality β, and, therefore, so does $\mathbf{a}^*$.

Essentially, each value a_k^* in Equations 4 and 5 is a probability (or scaled weight) measuring the effect of each $\mathbf{x}_k$ for image classification. Now we are ready to weigh each RoI using $\mathbf{a}^*$. In Equation 6 each RoI vector $\mathbf{x}_k$ is multiplied by the corresponding 'probability' a_k^* to obtain feature vector $\mathbf{c}$. In order to do this, we transpose $\mathbf{X}$, so that a_k^* is multiplied by each channel (feature) in the corresponding k^{th} RoI; hence, vector $\mathbf{c}$ in Equation 6 has dimensionality C.

Although in the context of RoI weighing, vector of features $\mathbf{c}$ can be used as an output of the Attention layer, we followed the approach in [10]: we filtered $\mathbf{c}$ through another fully connected layer to get $\mathbf{c}^*$, Equation 7, and, finally, summed $\mathbf{z}$ and $\mathbf{c}^*$ elementwise, Equation 8: $\oplus$ is an elementwise operator to obtain vector $\mathbf{o}$, the final output of the Attention layer that expresses useful features extracted from RoIs.

D. LSTM with Attention Model

We attempt two variants of LSTM+Attention mechanism: single LSTM branch with attention (LSTM-1) that outputs a single feature vector and uses a single image classifier, like in the Attention above. The second approach, LSTM-2, uses two parallel LSTM+Attention branches: one for class-relevant RoIs and one for class-irrelevant RoIs, so Attention layer outputs

two feature vectors. LSTM with Attention is shown in Figure 3.

The input in the Attention layer is the same, $\mathbf{X}$ with the same dimensionality, $\beta \times C$, which is the input in the LSTM model, Equation 9. This is the first difference from the base model, as the batch $\mathbf{X}$ is not reshaped. Therefore, the dimensionality of the LSTM input sequence is $\beta \times C$, and, as explained earlier, RoIs are ordered by Euclidean distance from the origin.

The second important difference from the Attention model is the dimensionality of the hidden features in LSTM, C^*, which can be different to C. LSTM outputs two tensors: $\mathbf{H}$, the full history of the hidden features with dimensions $\beta \times C^* (C^*$ is a hyperparameter), and the last hidden output, $\mathbf{h}_\beta$ with C^* dimensions, Equation 9. Each row in $\mathbf{H}$ is the feature outputs of the corresponding hidden layer, $\mathbf{h}_k$, Equation 10.

To get a better expression, $\mathbf{h}_\beta$ is filtered through a fully connected layer to get another feature vector $\mathbf{v}$, Equation 11. Then, we take a matrix-vector product of $\mathbf{H}$ and $\mathbf{v}$ to obtain a vector of raw features $\mathbf{b}$, Equation 12. This is yet another important difference from the Attention model: raw features are computed for ordered LSTM hidden features, rather than RoIs, from which they were extracted. Raw feature vector is transformed into softmax probability, b_k^*, Equation 13, and $\mathbf{b}^*$ is the vector of the softmax distribution, Equation 14.

We take matrix-vector product of LSTM history, $\mathbf{H}$ and $\mathbf{b}^*$ to get feature vector $\mathbf{d}$. This is another important difference from the Attention model, becasue softmax distribution scales LSTM hidden features, $\mathbf{h}_k$, rather than RoIs (see Equation 6). After another fully connected filter, Equation 16, feature vector $\mathbf{d}^*$ is summed elementwise with feature vector $\mathbf{v}$ from Equation 11 to output feature vector $\mathbf{p}$, Equation 17.

Attention and LSTM-1 output a single vector of features, respectively $\mathbf{o}$ or $\mathbf{p}$, into the image classifier. Instead, LSTM-2 outputs two vectors from two different LSTM+Attention layers: $\mathbf{p_1}, \mathbf{p_2}$ - class-relevant (positive), and class-irrelevant(negative) features. The architecture in these two layers is identical to LSTM-1, and the computation in both layers is done using Equation 9-17. These outputs are used as an input in the final stage of the model, image classifier, which has the same architecture as in [14].

$$\mathbf{H}, \mathbf{h}_\beta = LSTM(\mathbf{X}) \tag{9}$$
$$\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2} \dots \mathbf{h}_\beta] \tag{10}$$
$$\mathbf{v} = Linear(\mathbf{h}_\beta) \tag{11}$$
$$\mathbf{b} = \mathbf{H}\mathbf{v} \tag{12}$$
$$b_k^* = \frac{e^{b_k}}{\sum_{k=1}^{\beta} e^{b_k}} \tag{13}$$
$$\mathbf{b}^* = [b'_1, b'_2 \dots b'_\beta] \tag{14}$$
$$\mathbf{d} = \mathbf{H^T}\mathbf{b}^* \tag{15}$$
$$\mathbf{d}^* = Linear(\mathbf{d}) \tag{16}$$
$$\mathbf{p} = \mathbf{d}^* \oplus \mathbf{v} \tag{17}$$

IV. Experimental Setup

We compare the new model to One Shot Model with Affinity from [14] for both problems. For the classification problem, we also compare it to COVID-CT-Mask-Net, [13], and a suite of the state-of-the-art benchmark models. For the segmentation problem, we compare it to Mask R-CNN. In all experiments we used Adam optimizer with a learning rate of $1e-5$ and weight decay of $1e-3$.

All input images are scaled to 512×512, the dimensionality of all RoIs (mask features) is $256 \times 28 \times 28$, batch size β is set to 16, N is set to 1 (larger values did not improve the results, but slowed down the training). Matrix $\mathbf{X}$ has dimensionality 16×256. C^* is set to 256, $\mathbf{H}$ has dimensionality 16×256. Other hyperparameters of Mask R-CNN and One Shot Model are the same as in [14].

A. Segmentation Results

We use MS COCO 2017 main criterion, mean Average Precision (mAP) AP@[0.5:0.95:0.05]IoU, and two Intersect over Union (IoU) thresholds: AP@50%IoU and AP@75%IoU, see [7] for the details and [13] for the previous implementations in the context of COVID-19.

Segmentation results on the test split are reported in Table I. Attention model with ResNet50 feature extractor achieves the highest mAP of 0.4469, thus outperforming the highest scoring Mask R-CNN model (also with ResNet50 feature extractor and 5 FPN layers) by 0.0594, next-best Attention model with ResNet34 backbone by 0.0077 and One Shot Model with Affinity from [14] by 0.0226. It also achieves top precision with AP@75%IoU criterion, 0.4423. Attention model with ResNet34 feature extractor achieves top precision for AP@50%IoU criterion, 0.6405.

TABLE I: Average Precision on the segmentation test split (100 images). Best results in bold.

	Model	Model size	AP@0.5 IoU	AP@0.75 IoU	AP@[0.5:0.95]IoU
Attention	ResNet18	23M	0.5670	0.4201	0.4018
	ResNet34	33M	**0.6405**	0.4350	0.4392
	ResNet50	35M	0.6350	**0.4423**	**0.4469**
	ResNeXt50	35M	0.5364	0.4087	0.3959
	ResNeXt101	99M	0.5879	0.4226	0.4118
	One Shot Model [14]	36M	0.5903	0.3891	0.4242
	Mask R-CNN	44M	0.5026	0.4194	0.3875
	Mask R-CNN (heads only)	44M	0.4442	0.3791	0.3354

B. Classification Results

Accuracy of the model is computed using sensitivity/recall per class and class-adjusted F1 score for the overall model. In our implementation of F1 score, the weights (shares) of classes in the test set are taken into consideration. In many publications, COVID-19 sensitivity is considered to be a particularly important measure.

Results in Table II demonstrate that Attention model with ResNet50 backbone confidently outperforms all other models across all accuracy metrics.

For COVID-19 sensitivity, it improves on the next best model, Attention with ResNet34 backbone by 5.69%, One Shot Model with Affinity by 1.96%, COVID-CT-Mask-Net by 4.51%, and the best benchmark model (DenseNet121) by 2.57%, and the lowest-scoring one, ResNet34, by 6.61%.

For CP, the same values are 0.90% (ResNet34), 2.98% (One Shot Model+Affinity), 6.92% (COVID-CT-Mask-Net), 0.90% (ResNet50) and 10.09%(ResNeXt50).

For the Normal class these values are 0.01%(ResNet34), 3.57% (One Shot Model+Affinity), 8.11% (COVID-CT-Mask-Net), 0.01%(ResNet34), and 14.90%(ResNeXt50).

Finally, for F1 score, these values are 1.31% (ResNet50), 3.03% (One Shot Model+Affinity), 6.68% (COVID-CT-Mask-Net), 1.31% (ResNet50) and 10.87% (ReNeXt50).

Overall, Attention model with ResNet50+FPN backbone achieves best results across all problems, except segmentation AP@50%IoU, in which ResNet34 improves on it by 0.005.

C. Ablation Studies

In Section III-D we presented two extensions to the Attention model's architecture for sequential RoI input, LSTM-1 and LSTM-2. Also, as explained in Sections II-A and III-D, the order of inputs in LSTM is determined by the RoI's rank in the batch sorted by the RoIs' Euclidean distance from the origin. Hyperparameters, including RoI layer and Attention functionality, for both LSTM models were the same as for the Attention model. Model sizes in Tables III and IV show that LSTM layers add only a small overhead to the base model.

1) Segmentation results: We use the same MS COCO metrics to compare models as in Section IV-A. Results of ablation experiments are reported in Table III and Figures 4a-4c. Barcharts in Figure 4 show the LSTM models' performance compared to the base model for the same backbone architecture.

Two LSTM-2 architectures clearly stand out: LSTM-2 with ResNet18 backbone and LSTM-2 with ResNeXt101 backbone, as they confidently outperform both LSTM-1 and base models. by a large margin, including mAP, main MS COCO criterion across architectures. LSTM-2 improves base model by 0.056 and LSTM-1 by 0.069.

Overall, for the mAP metric, LSTM-2 performs best across all feature extractor architectures, except ResNet50, where its precision is 0.001 lower than base model. LSTM-1 with ResNet34 and ResNext50 backbones also outperform Attention model, albeit with a lower margin. Both LSTM-2 top models outperform LSTM-1 top models for the same architecture across all metrics (ResNet18 and ResNeXt101 bars in Figures 4a-4c).

Results in Table III confirm these findings. For mAP and AP@75%IoU, LSTM-2 with ResNext101 backbone achieves the highest accuracy: 0.4683 for mAP and 0.4891 for AP@75%IoU. LSTM-1 with ResNet34 backbone achieves the highest accuracy for AP@50%IoU metric.

For the AP@50%IoU criterion, top LSTM-1 result (ResNet34) outperforms top base model (ResNet34) result by 0.002. For AP@75%IoU metric top Attention model (ResNet50) is outperformed a number of models. Top model, LSTM-2 (ResNeXt101) outperforms it by 0.0467, and top LSTM-1 (ResNet34) by 0.0401.

Finally, for mAP metric top base model (ResNet50) is

TABLE II: Accuracy results on the COVIDx-CT test split (21191 images). Per-class sensitivity, overall and $F1$ scores are reported. Best results in bold.

	Model	Model size	COVID-19	CP	Negative	F_1 score
Attention	ResNet18	23M	90.34%	94.96%	98.58%	95.63%
	ResNet34	33M	94.75%	92.66%	89.68%	91.80%
	ResNet50	35M	**95.32%**	**98.55%**	**99.21%**	**98.19%**
	ResNext50	35M	88.96%	93.55%	95.68%	93.66%
	ResNext101	99M	91.71%	96.83%	97.27%	96.00%
	One Shot Model [14]	36M	93.35%	95.56%	95.63%	95.16%
	COVID-CT-Mask-Net [13]	32M	90.80%	91.62%	91.10%	91.50%
	ResNet18	11M	92.59%	96.25%	92.03%	93.61%
	ResNet34	21M	88.70%	96.66%	99.20%	96.17%
	ResNet50	25M	91.04%	97.64%	98.97%	96.88%
	ResNeXt50	25M	91.94%	88.45%	84.30%	87.31%
	ResNeXt101	88M	91.58%	92.13%	94.02%	92.86%
	DenseNet121	8M	92.64%	96.16%	98.98%	96.69%
	DenseNet169	14M	89.37%	96.78%	98.12%	95.86%

also outperformed by both LSTM models. Top LSTM-2 model (ReNeXt101) improves on base model by 0.0213, and top LSTM-1 model (ResNet34) by 0.0030.

At the same time, for AP@50%IoU none of the LSTM-2 models achieves top-3 results. For AP@75%IoU, LSTM-2 achieves the best and third-best results (ResNeXt101 and ResNet50), as LSTM-1 (ResNet34) achieves the second-best one. For mAP, LSTM-2 (ResNeXt101) achieves the top result, LSTM-1 (ResNet34) second-best and base model (ResNet50) third-best.

2) Classification results: Classification results for the same setup are reported in Table IV and Figures 5a-5d. As reported in Figures 5a-5d, three LSTM-2 models: ResNet18, ReNeXt50 and ResNeXt101 backbones outperform both base model and LSTM-1 across all 4 metrics for their respective architecture. On top of that, LSTM-2 with ResNet34 outperforms base model only. LSTM-1 with ResNet34 outperform base model and LSTM-2 on 3 out of 4 metrics (except Common Pneumonia), in which it lags behind LSTM-2 only by 0.02%.

In Table IV, LSTM-2 with 3 different architectures achieve 3 out of 4 top results. For COVID-19, LSTM-2 with ResNeXt101 backbone gets 95.74% sensitivity, for CP LSTM-2 with ResNet34 gets 98.91% sensitivity, and for Normal, LSTM-2 with ReNeXt50 achieves 99.77% sensitivity.

Top F1 score, 98.56% is achieved by LSTM-1 with ResNet34 backbone. For COVID-19, top LSTM-1 result, ResNet34 improves on baseline by 0.14%, and top LSTM-2 result improves it by 0.42%. For CP, top LSTM-1 result (ResNet34) outperforms base model (98.55%) by 0.34% and top LSTM-2 result outperforms it by 0.35%.

For Normal class, top base result is 99.21%, LSTM-1 improves it by 0.49% and LSTM-2 by 0.56%. For F1 score top base result is 98.19%, improved by the top LSTM-1 result by 0.37% and top LSTM-2 result by 0.17%.

As reported in Table IV, for COVID-19 sensitivity, LSTM-2 with ResNeXt101 and ResNeXt50 achieve best and second-best results, and LSTM-1 third best. For CP, LSTM-2 achieves the best result (ResNeXt101), LSTM-1 second-best (ResNet34), and Attention model third-best (ResNet50). For Normal, LSTM-2 achieves the best and third-best results (ResNeXt50 and ResNet34), and LSTM-1 second best (ResNet34). For F1 score, LSTM-2 achieves the second- and third-best results (ResNet34 and ResNeXt50), and LSTM-1 the best one (ResNet34).

Overall, across both problems and metrics, LSTM-2 with ResNeXt101 backbone achieves top results in 3 categories (COVID-19 sensitivity, AP@75%IoU and mAP). LSTM-1 with ResNet34 achieves top results in two categories (AP@50%IoU and F1 score), LSTM-2 with ResNet34 in one category (CP), and LSTM-2 with ResNeXt50 also in one catogory (Normal). Therefore, LSTM-2 with different backbones achieves 5 top results out of a total of 7.

Considering top 3 results for each category, LSTM-2 achieved 5 top results (AP@75%IoU, mAP, COVID-19, CP, Normal), 2 second-best (COVID-19, F1 score) and 3 third-best (AP@75%IoU, Normal, F1 score). LSTM-1 achieved two top results (AP@50%IoU, F1 score), 4 second-best (AP@75%IoU, mAP, CP, Normal) and one third-best (COVID-19). Attention model achieved one second-best result (AP@50%IoU) and three third-best (AP@50%IoU, mAP, CP).

Another important result from the ablation study is that the architecture, depth or size (number of parameters) do not determine the model's accuracy. For example, LSTM-1 with ResNeXt50 (34M parameters) vs ResNeXt101 (98M

TABLE III: Results for ablation experiments on the segmentation test data. Bold: best result for this backbone architecture, blue underline: best result for the metric, green:second-best, black: third-best.

Backbone	Architecture	Model Size	AP@0.5 IoU	AP@0.75 IoU	AP@[0.5:0.95]IoU
ResNet18	Attention model	23M	0.5670	0.4201	0.4018
	LSTM-1	22M	0.5398	0.4250	0.3911
	LSTM-2	23M	**0.6146**	**0.4390**	**0.4229**
ResNet34	Attention model	33M	0.6405	0.4350	0.4392
	LSTM-1	32M	**0.6434**	0.4825	**0.4472**
	LSTM-2	33M	0.6288	**0.4357**	0.4457
ResNet50	Attention model	35M	**0.6350**	0.4423	**0.4469**
	LSTM-1	35M	0.5961	0.4178	0.4137
	LSTM-2	36M	0.6234	**0.4561**	0.4453
ResNeXt50	Attention model	35M	0.5364	0.4087	0.3959
	LSTM-1	34M	0.5761	**0.4112**	0.3981
	LSTM-2	36M	**0.6269**	0.4048	**0.4153**
ResNeXt101	Attention model	99M	0.5879	0.4226	0.4118
	LSTM-1	98M	0.5591	0.4203	0.3983
	LSTM-2	99M	**0.6187**	**0.4891**	**0.4683**

TABLE IV: Results for ablation experiments on the classification test data. Bold: best result for this backbone architecture, blue underline: best result for the metric, green:second-best, black: third-best

Backbone	Architecture	Model size	COVID-19	CP	Normal	F1 score
ResNet18	Attention model	23M	90.34%	94.96%	98.58%	95.63%
	LSTM-1	22M	92.04%	88.98%	97.81%	93.59%
	LSTM-2	23M	**92.08%**	**98.05%**	**99.25%**	**97.37%**
ResNet34	Attention model	33M	94.75%	92.66%	89.68%	91.80%
	LSTM-1	32M	**95.46%**	98.89%	**99.70%**	**98.56%**
	LSTM-2	33M	95.16%	**98.91%**	99.37%	98.36%
ResNet50	Attention model	35M	**95.32%**	**98.55%**	99.21%	**98.19%**
	LSTM-1	35M	93.62%	95.88%	**99.24%**	96.93%
	LSTM-2	36M	94.45%	96.91%	98.73%	97.22%
ResNeXt50	Attention model	35M	88.96%	93.55%	95.86%	93.66%
	LSTM-1	34M	89.48%	93.07%	99.22%	95.07%
	LSTM-2	36M	**95.58%**	**97.86%**	**99.77%**	**98.25%**
ResNeXt101	Attention model	99M	91.71%	96.83%	97.27%	96.00%
	LSTM-1	98M	82.88%	91.31%	92.18%	90.04%
	LSTM-2	99M	**95.74%**	**98.13%**	**99.27%**	**98.15%**

parameters) leads to a large drop across all classification accuracy criteria, i.e. a smaller model outperforms a much larger one. At the same time, for ResNeXt50 architecture. LSTM-2 has about 1.5M parameters more that either base of LSTM-1. Nevertheless, it confidently outperforms both of them across all classification criteria.

V. Limitations of the methodology

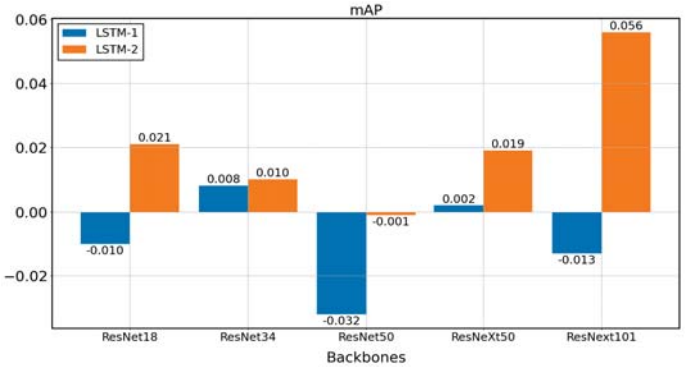
In Sections I and II, we mentioned the problem of transferability (generalization), or domain adaptation of the OS COVID-19 models to other datasets and their implementation in the real hospital environment. To the best of our knowledge, no known OS solution, trained on one dataset, was then successfully evaluated on an entirely different one out-of-the-box, or implemented in the real-life medical facility. We do admit though that such proprietary solutions may exist though.

A detailed discussion of this situation (up until October 2020) is presented in [20]: a large number of methodological flaws, lack of information about hyperparameters and architectures, and unavailable datasets prevent the replication, fair comparison, generalization and real-life implementation of the models. Unlike benchmark datasets, like MS COCO or Pascal VOC, on which general-purpose deep learning models can be trained, evaluated and compared, COVID-19 datasets for both classification and segmentation problems are yet to be developed.

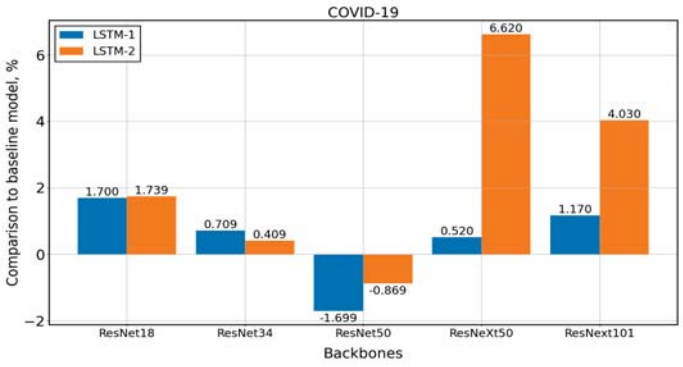
Although we do not adapt or generalize the presented models, they have a strong potential for real-life applications given their advanced architecture, inherited from Mask R-CNN and a small training data.

VI. Conclusions

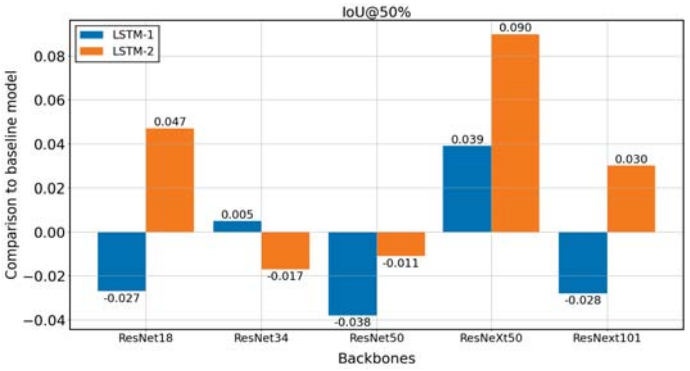
In this paper we presented a novel methodology that combines a lesion detection and CT scan slices classification



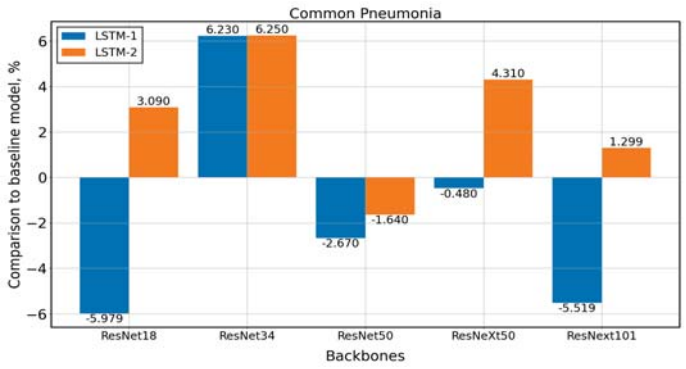
(a) mAP



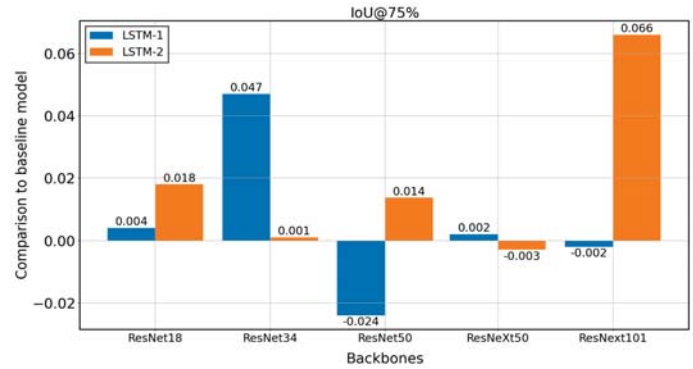
(a) COVID-19



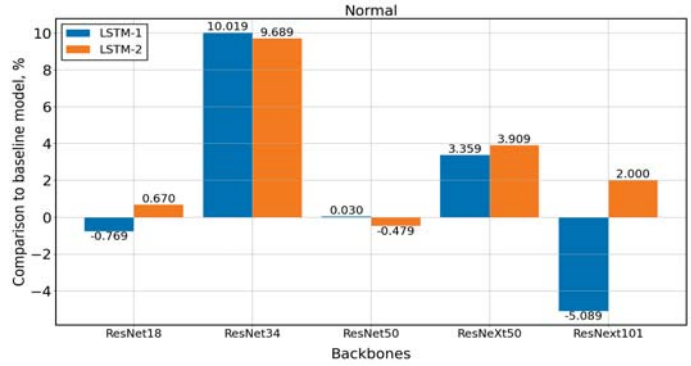
(b) AP@50%IoU



(b) Common Pneumonia



(c) AP@75%IoU

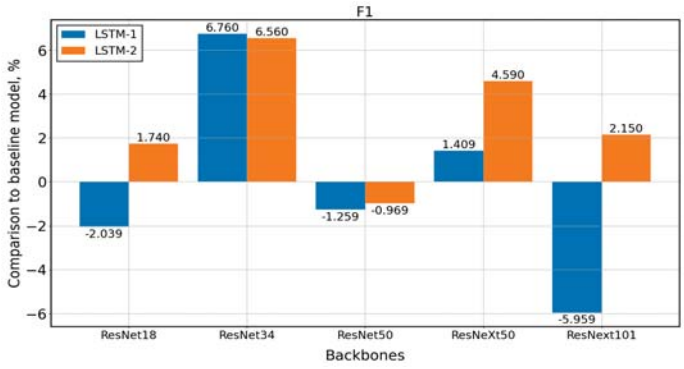


(c) Normal

Fig. 4: Comparison of lesion segmentation precision of LSTM-1 and LSTM-2 to Attention model.

model with Attention mechanism and Long Short-Term Memory Net to explore relationship among Regions of Interest (expressed through mask features) to segment lesions and classify chest CT scans.

Our base model with ResNet50+FPN backbone and Attention mechanism on Regions of Interest achieves 0.4469 mean average precision, 95.32% COVID-19 sensitivity and 98.19% F1 score, outperforming both Mask R-CNN



(d) F1 score

Fig. 5: Comparison of classification accuracy of LSTM-1 and LSTM-2 models to Attention model, in %.

(segmentation) and a suite of benchmark models (classification).

We ran a set of ablation studies, by adding either one or two LSTM layers with Attention. The model with ResNeXt101+FPN backbone and two LSTM branches, achieved 0.4683 mean average precision, 95.74% COVID-19 sensitivity and 98.15% F_1 score, the model with a single LSTM layer and ResNet34 backbone achieved 0.4472 mAP, 95.46% COVID-19 sensitivity and 98.56% F1 score. Both of them improve on results achieved by the Attention model.

VII. Acknowledgements

The authors would like to thank Dr Esther Mondragon, Professor Eduardo Alonso and Dr Giacomo Tarrone for their valuable advice and recommendations that helped improve the quality of the paper.

Aram Ter-Sarkisov is a Lecturer at City, University of London. For several years he has been working on the application of Deep Learning in Computer Vision, and his most recent work is on the segmentation of lesions and classification of chest CT scans for COVID-19 prediction. He earned his MSc in Statistics from the University of Auckland, New Zealand and PhD in Computer Science from Massey University, New Zealand. Email:alex.ter-sarkisov@city.ac.uk

References

[1] Y.-H. Wu, S.-H. Gao, J. Mei, J. Xu, D.-P. Fan, C.-W. Zhao, and M.-M. Cheng, "Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation," *arXiv preprint arXiv:2004.07054*, 2020.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[4] J. Zhao, Y. Zhang, X. He, and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865*, 2020.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images," *Informatics in medicine unlocked*, vol. 20, p. 100412, 2020.

[10] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[11] S. Yazdani, S. Minaee, R. Kafieh, N. Saeedizadeh, and M. Sonka, "Covid ct-net: Predicting covid-19 from chest ct images using attentional convolutional network," *arXiv preprint arXiv:2009.05096*, 2020.

[12] J. Wang, Y. Bao, Y. Wen, H. Lu, H. Luo, Y. Xiang, X. Li, C. Liu, and D. Qian, "Prior-attention residual learning for more discriminative covid-19 screening in ct images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2572–2583, 2020.

[13] A. Ter-Sarkisov, "COVID-CT-Mask-Net: Prediction of COVID-19 from CT Scans Using Regional Features," *medRxiv*, 2020. [Online]. Available: https://github.com/AlexTS1980/COVID-CT-Mask-Net

[14] ——, "One shot model for the prediction of covid-19 and lesions segmentation in chest ct scans through the affinity among lesion mask features," *medRxiv*, 2021.

[15] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3367–3375.

[16] Q. Yao and X. Gong, "Exploiting lstm for joint object and semantic part detection," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 498–512.

[17] J. H. Bappy, C. Simons, L. Nataraj, B. Manjunath, and A. K. Roy-Chowdhury, "Hybrid lstm and encoder–decoder architecture for detection of image forgeries," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3286–3300, 2019.

[18] H. Gunraj, L. Wang, and A. 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.

[19] X. Li, X. Fang, Y. Bian, and J. Lu, "Comparison of chest ct findings between covid-19 pneumonia and other types of viral pneumonia: a two-center retrospective study," *European radiology*, pp. 1–9, 2020.

[20] M. Roberts, D. Driggs, M. Thorpe, J. Gilbey, M. Yeung, S. Ursprung, A. I. Aviles-Rivero, C. Etmann, C. McCague, L. Beer *et al.*, "Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 199–217, 2021.