DeepBreath

Detecting emphysema presence in longitudinal CT scans of human lungs using convolutional and recurrent neural networks

Anders Geil*

Department of Computer Science
University of Copenhagen
<xkh299@alumni.ku.dk>

June 18, 2018

Abstract

We investigate the viability of combining convolutional and recurrent neural networks for the estimation of emphysema extent from annual CT scans of human lungs. First, we demonstrate how the GAPNet convolutional neural network architecture may be successfully implemented and trained for use with single CT scans. Second, using this model as a foundation, we show how a recurrent neural network may be adapted to achieve improved performance by utilizing multiple, unlabelled CT scans. We achieve an overall accuracy of 69% *across all emphysema grades for the convolutional neural network and are able to distinguish between healthy and unhealthy subjects with an accuracy of* 80%*. Extending the model with a recurrent module, we are able to increase the overall accuracy to* 74%*, with* 81% *of subjects correctly declared as either healthy or unhealthy. Finally, our results appear to suggest that the extended model may be utilizing temporal dependencies to successfully accommodate underestimation biases inherent in the underlying convolutional model.*

I. Introduction

Emphysema is a lung disease causing shortness of breath due to the progressive destruction of lung tissue. Over time, the inner walls of the alveoli (air sacs) over-inflate and rupture, obstructing the exchange of oxygen and carbon dioxide in the lungs. Traditionally, the condition is diagnosed by experts through manual, visual inspection of the lung CT scans. While both time-consuming and costly, this procedure further suffers from significant inter-rater variability. To alleviate these issues, this article contributes to the recent research on utilizing machine learning to automatically detect emphysema from CT scans of human lungs.

Specifically, this paper investigates the viability of incorporating a temporal dimension in established models predicting the extent of emphysema prevalence. By learning the temporal dependencies inherent in prior CT scans, the models may become able to exploit the progressive nature of the condition. In turn, by utilizing previously performed CT scans, the model may leverage these temporal developments to deliver more accurate and consistent estimates of the current emphysema extent when compared to estimates performed by existing cross-sectional models.

II. Background

This section briefly reviews the existing literature on automated analysis of video sequences and its most relevant applications within medical image analysis, respectively.

i. Video analysis

Modelling temporal dependencies across distinct images has recently found great progress within the field of video analysis. The earli-

* Supervisors:
Silas Nyboe Ørting <silas@di.ku.dk>
Jens Petersen <phup@di.ku.dk>

est study, Ji et al. (2013), experiments with the use of stacked images and 3D convolutions to capture temporal developments for human action recognition within videos. While this architecture succesfully captures local temporal developments, the study resorts to feature engineering to capture the more global, long-term dependencies in a parallel pipeline. A similar two-stream architecture is found in Simonyan and Zisserman (2014), who also split the data processing pipeline into two parallel convolutional networks. First, a spatial stream CNN analyzes the contents of single RGB-frames, while, second, a temporal stream CNN utilizes optical flow to track long-term movement across multiple frames. Again, however, handcrafted filters are required to extract optical flow features, why none of the models satisfy the constraints of end-to-end modelling.

Karpathy et al. (2014) investigates the effects of fusing temporal information in CNN architectures. The study distinguishes between early fusion, where filters in the first convolutional layer immediately merge the temporal information in the input; late fusion, where the temporal information is maintained until the final, fully connected layers; and slow fusion, where the convolutional layers gradually collapse the temporal information through 3D convolutions. While early fusion is computationally more efficient, the network cannot fully utilize the temporal information of the input. Similarly, since late fusion is computationally demanding, only relatively distant frames are used as input, effectively disregarding most of the information present in the input. Slow fusion, then, is shown to provide the best compromise by merging distant frames first and adjacent frames later.

In a similar vein, Tran et al. (2015) explores the impact of varying temporal kernel depth in the 3D convolutional neural network proposed by Ji et al. (2013). Across multiple video-based datasets, the study finds a fixed temporal depth (3x3x3) to conveniently outperform both gradually increasing and decreasing temporal kernel depths along the network architecture.

Yao et al. (2015) draw a helpful theoretical distinction between local and global temporal structure. Using a 3D convolutional neural network to extract local temporal structure, the study further proposes the use of a recurrent neural network to capture global temporal structure by selectively attending to the various local motion descriptors provided by the CNN. In this way, the network is able to provide a natural language output string description of the entire video sequence.

In parallel with this study, Donahue et al. (2015) propose a more generalized long-term recurrent convolutional network (LRCN). The LRCN also combines a base CNN with a subsequent RNN to capture both local and global temporal structure. By parallelizing the CNN architecture, however, the LRCN distributes the computational burden of gradually fusing temporal information in a single 3D CNN across its individual time steps. Instead, the weights of a common 2D CNN are shared across multiple input frames and processed in parallel. By combining the features of temporally adjacent inputs, the consecutive RNN is then able to model both the local temporal structure in the given input as well as track the global temporal developments through its internal state. Finally, the study explores the effects of intermediate fully connected layers prior to the RNN, but the results suggest better performance for fewer dense layers, leaving the RNN less constrained in the process of dimensionality reduction before the final output.

A more experimental configuration is consequently proposed by Ballas et al. (2015), who integrate the convolutional component of the CNN directly into the linear product operations of the gated recurrent units (GRU) in the RNN. In this way, the modular structure of the LRCN is fully integrated, effectively escaping the architectural distinction between local and global temporal structure while theoretically reducing the total amount of parameters in the model.

In general, however, the performance benefits achieved by incorporating temporal dependencies in models for video analysis is limited to 1-4% over a basic 2D CNN. This may, how-

ever, be partially attributed to the nature of current video datasets (e.g. video sequences of sports), where the potential information gained by additional temporal movement is arguably rather limited over a sequence of individual frames. For this reason, the next section explores the applications of longitudinal modelling within the domain of medical image analysis.

ii. Medical image analysis

In the medical field, previous research on emphysema detection is composed solely of cross-sectional studies modelling the presence and extent of emphysema in a single CT scan at a given point in time. In particular, Bortsova et al. (2018) compare two different model architectures for multiple instance learning (MIL) and learning by label proportions (LLP) to predict emphysema presence and extent, respectively. Both models build on the same base convolutional neural network (CNN), adding on either a global average pooling layer to directly deduce the total proportion of emphysema (GAP-Net), or a hidden segmentation layer succeeded by a layer explicitly comparing the proportion of segmented emphysema to the given region mask. While the results generally indicate the latter outperforming the former overall, they both provide comparable results on medium to large datasets.

Although no longitudinal studies have been made on emphysema detection specifically, Litjens et al. (2017) provide a comprehensive review on the use of deep learning in medical image analysis at large. In fetal imaging, for instance, Baumgartner et al. (2016) used CNNs on individual frames to localize key scan planes of e.g. brain and spine positions in fetal ultrasound videos. Chen et al. (2015) extended this approach by employing LSTM models to incorporate the temporal information inherent in video sequences. Further, within cardiac imaging, Poudel et al. (2016) combined the common U-net CNN architecture with a recurrent neural network (RNN) consisting of gated recurrent units (GRU) to segment the left ventricle slice by slice, continuously utilizing information from the previous slices when segmenting the next one. Similarly, Kong et al. (2016) combined a 2D CNN with a RNN based on long short-term memory (LSTM) units to detect end-diastole and end-systole frames in temporal MRI data of the heart. In lung imaging, only Ypsilantis and Montana (2016) utilized the temporal dimension by combining a CNN and RNN to detect pulmonary nodules in lung CT scans, providing a 9% increase in prediction sensitivity over a base CNN.

III. Methods

This section first describes the data used for modelling. Convolutional neural networks are then briefly introduced to present the base model, GAPNet. Finally, recurrent neural networks are explained and incorporated into an extended model. A visualization of the resulting two model architectures can be found in figures 1 and 2, respectively.

i. Data

The Danish Lung Cancer Screening Trial (DLCST) is a randomized controlled trial with 4,104 participants. Participants, aged 50-70 years, were either current or previous smokers with a smoking history of at least 20 pack years[1] and a forced expiratory volume in 1 second (FEV_1) of at least 30% during the recruitment period from 2004 to 2006 (Pedersen et al., 2009). The control group comprised 2,052 participants, while the remaining 2,052 participants were screened using low-dose thoracic computed tomography (CT). A total of 1,388 participants were screened annually for the full five consecutive years of the study period. During screening, participants were asked to hyperventilate thrice and then inhale maximally, holding their breath during imaging (Wille et al., 2014, 2693).

Wille et al. (2014) performed visual inspections of the first and last available CT scans from each participant. The two observers, both MDs and PhD students with backgrounds in

[1] A pack year is defined as smoking 20 cigarettes (one pack) per day for one year.

	0%	1-5%	6-25%	26-50%	51-75%	76-100%
All	982 (70.7%)	225 (16.2%)	128 (9.2%)	40 (2.9%)	11 (0.8%)	3 (0.2%)
Training	563 (72.3%)	120 (15.4%)	65 (8.3%)	23 (2.9%)	6 (0.8%)	2 (0.3%)
Validation	227 (68.2%)	62 (18.6%)	33 (9.9%)	8 (2.4%)	3 (0.9%)	0 (0.0%)
Test	191 (69.2%)	43 (15.6%)	30 (10.9%)	9 (3.3%)	2 (0.7%)	1 (0.4%)
Reduced training	70 (70%)	17 (17%)	9 (9%)	4 (4%)	0 (0%)	0 (0%)

Table 1: *Number (proportion) of samples by emphysema grade across data partitions.*

pulmonology and radiology, were blinded to participant identity, clinical data, and examination date during evaluation. The visual assessment was performed individually on each of three lung regions: upper zone (above carina), middle zone (between carina and inferior pulmonary vein), and lower zone (below inferior pulmonary vein) (Wille et al., 2014, 2694). For each region, the extent of emphysema was assigned to one of the intervals: 0%, 1-5%, 6-25%, 26-50%, 51-75%, 76-100%.

In this study, we focus exclusively on the upper right lung region, which demonstrates the least interobserver variability. Only participants who were scanned for five consecutive years are included, and we use only the emphysema grade for the last CT scan during training. Table 1 shows the strong imbalance of the dataset. The vast majority (70.7%) of participants are healthy with an estimated emphysema extent of 0%. A minority of participants (29.3%) display some signs of emphysema with larger extents being increasingly less common. Approximately 20% of the full dataset is randomly selected and reserved for testing, while the remaining data is randomly split into a training (70%) and validation set (30%). The resulting splits roughly reflect the relative imbalances of the full dataset (see table 1).

ii. Convolutional Neural Networks

CT scans are stored as 3-dimensional arrays with voxel values representing the attenuation of the corresponding tissue. Since the underlying tissue has structure, neighbouring voxels are not independent. Convolutional neural networks take advantage of this fact by utilizing the spatial information present in the input image. Locally connected kernels (aka filters), with a limited receptive field, learn local spatial structure by convolving with a small region of the input image. By sliding kernels over every region of the input image and sharing the kernel weights throughout, convolutional neural networks are able to identify similar features in different image regions and become translationally invariant.

In turn, the output of these convolutions are stored in feature maps (aka activation maps), which may represent local microstructures (e.g. textures) in the input image. By continuously convolving over these feature maps, kernels in subsequent convolutional layers gradually achieve a broader receptive field of the original input image. Through the combination of local structures identified in lower layers, convolutional networks obtain gradually more abstract representations of the image structure. In this way, convolutional neural networks may become able to distinguish healthy lung tissue from emphysema by modelling the spatial properties of their structural differences in texture.

GAPNet

We adopt the convolutional neural network architecture used by Bortsova et al. (2018) as our baseline model since this model has previously been shown to successfully adapt to the task of emphysema detection through CT scans. GAPNet, short for global average pooling network, takes a 3D image of a lung region as input, converts it to a set of 3D feature maps, pools the features maps using global average pooling, and then combines the averages into a prediction of the proportion of emphysema present in the given input image.

GAPNet comprises a total of 47 layers orga-

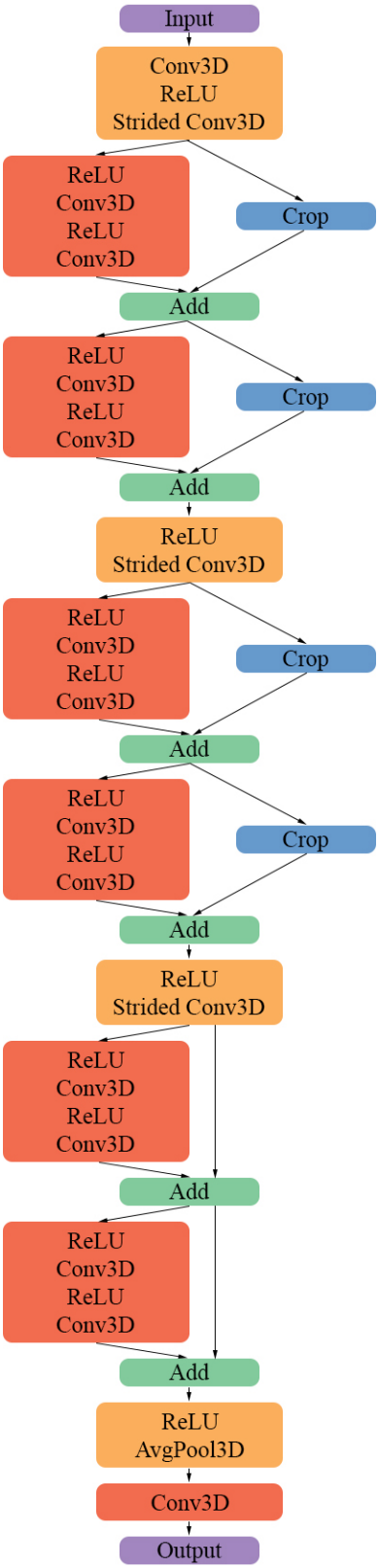

Figure 1: *Visualization of the base model (GAPNet).*

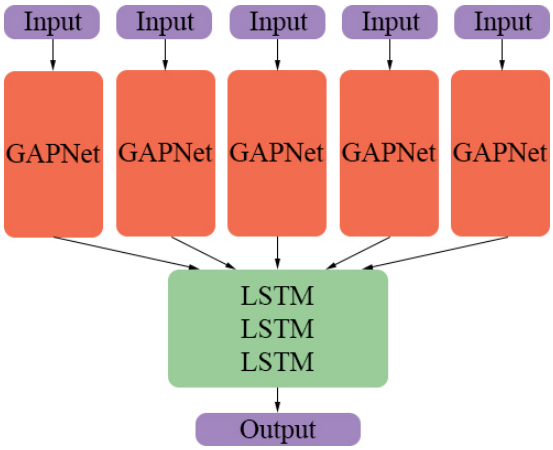

Figure 2: *Visualization of the extended model. GAPNet weights are shared.*

nized into a sequence of blocks. Each block consists of an alternating series of convolutional layers and rectified linear units (ReLU) affording nonlinearity by thresholding the signal at zero. Between each such block of convolutions, an intermediate, strided convolutional layer gradually compresses the learned features by reducing the input dimensionality after each block by half. This essentially forces the network to distill only the most important features before entering the next block.

Due to the sheer depth of the network, skip connections, routing information around each residual block of convolutions, are introduced in an attempt to counter potential issues with vanishing gradients. Since these connections copy the signal prior to the convolutional block and additively merge it back to the output of each block, the error signal is split between the skip connection and the convolutional block during backpropagation. In this way, the skip connections effectively helps preserve a constant error signal throughout the model by circumventing the use of repeated weight multiplication via use of the chain rule.

Additionally, skip connections allow deeper layers, typically generating more abstract features, to directly draw on representations from lower level layers, typically representing more concrete features. In this way, the model may dynamically combine low- and high-level fea-

tures throughout the network, irrespective of the activation maps generated within the individual, convolutional blocks.

iii. Recurrent Neural Networks

Extending our baseline network to utilize temporal information, we draw on recurrent neural networks. Recurrent neural networks incorporate self-referencing inner cells between their input and output. These loops allow information to persist within the cells of the network over time. By unraveling the loops, recurrent neural networks can be viewed as the same network continuously passing information to a future version of itself. In addition to the external input, the loop effectively provides a second, internal input reflecting the state of previous cells.

Recurrent neural networks are therefore sequential in nature, and we may interpret each feature vector input to the network, as it passes its inner cell state onwards, as a single time step in a sequence of temporally related inputs. Similarly, we may perceive the corresponding output at a given time step as influenced by information from previous time steps. In this way, our model achieves a sense of temporality as multiple, temporally related inputs potentially influence the predictions of their successors, i.e. by modelling their temporal dependencies.

To understand how these temporal dependencies are modelled, we will need to delve further into the architecture of the recurrent units. Specifically, we will be utilizing long short-term memory (LSTM) units, since these units have been shown to persistently propagate a constant error flow, efficiently reducing the risk of vanishing gradient problems (Hochreiter and Schmidhuber, 1997). Similar to the base model, this is also a common problem for recurrent neural networks, where excessive horizontal depth (across time steps) may cause gradients to decrease exponentially through the use of multiplication in the chain rule during backpropagation. Long short-term memory units specifically address this issue in the design of its inner cell architecture by maintaining both a cell and a hidden state.

LSTM

Long short-term memory units are designed around the maintenance of both an internal cell state, c_t, and a hidden state, h_t. The cell state may flow internally through all LSTM units, but cannot be directly accessed externally. Instead, a series of gates control the extent to which current information is forgotten and new information is added to the internal cell state. The information to be added, the candidate cell state, reflects both the direct input for the current time step and the hidden state of the previous time step. The hidden state, constituting a filtered representation of the cell state at a given time step, is passed locally between each pair of neighbouring units and may additionally be output directly. We will be covering these steps in more detail below.

Maintaining the internal cell state, c_t, the LSTM initially needs to decide how much of the information in the previous cell state, c_{t-1}, should be forgotten. This is achieved by the forget gate (1) through a weighting of the input feature vector, x_t, and the previous hidden state h_{t-1} using the learned weight matrix W_f and bias vector b_t. Since the sigmoid activation function compresses the output of the forget gate into the range $[0; 1]$, we may interpret the values as the relative proportion of each entry in c_{t-1} is to be retained in the new state c_t (0 to forget, 1 to retain). In a similar fashion, the input gate (2) determines the extent to which new information may flow into the cell state.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2}$$

The candidate cell, c'_t, represents the new information which may be added to the cell state through the input gate. The candidate cell values reflect a non-linear transformation of the current input, x_t, and the previous hidden state, h_{t-1}, compressed to the interval range $[-1; 1]$ by the hyperbolic tangent activation function (3). In this way, the resulting cell state values may be increased or decreased to avoid both under- and overshooting through the dynamic

adjustment of candidate cell values. In effect, cell states are simply updated by filtering the previous cell state with the forget gate, and adding the input gated candidate cell values (4).

$$c'_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ c'_t \quad (4)$$

Note that the previous cell state may be retained or replaced entirely, depending on the values provided by the forget and input gates. Using the internal cell state, LSTM's are able to retain information from previous inputs to emulate long term memory. On a related note, if the forget gate blocks all input, the cell state may only be modified by addition through the input gate. During backpropagation, the gradient error signal will then be distributed without multiplication through the chain rule. Thus, by maintaining this inner cell state, bypassing most of the computational logic, LSTM's are less susceptible to vanishing gradient problems.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$
$$h_t = o_t \circ \tanh(c_t) \quad (6)$$

Finally, the output gate, o_t, (5) is used to filter the amount of information passed on from the current cell state, c_t, to the hidden state, h_t. The hidden state, h_t, then represents a filtered version of the updated cell state, c_t, given the current input, x_t, and the previous hidden state, h_{t-1} (6). While the internal cell state may be perceived of as a long-term memory, representing the full sequence of units, the hidden state may then be vaguely thought of as a short-term memory, representing only a selection of features from immediately prior units. These hidden state features may additionally be output to the next layer as a representation of the current time step. Consequently, the hidden state corresponding to the last time step may also be output as a representation of the full sequence, indirectly reflecting all prior temporal dependencies.

iv. Overview

In summary, we will be drawing on the data source provided by Wille et al. (2014). First, we implement the convolutional neural network, GAPNet from Bortsova et al. (2018), as our base model. Second, in the extended model, we then replicate this network for each of the patients' five CT scans, sharing the weights across each copy, and model their temporal dependencies by adding three LSTM layers on top of the base models. Due to natural variance in participants' inhalations across the annual scans, we thus seek to model global temporal structure in correspondence with previous research approaches (see section II).

For the extended model, we remove the final convolutional layer in GAPNet, feeding the recurrent module five feature vectors rather than five prediction scores as input. In this way, we offer the recurrent module as many features as possible, while still retaining the key element of GAPNet, the global average pooling layer. Although subject to further experimentation, using three LSTM layers in the recurrent module reflects a trade-off between permitting end-to-end training of the full extended model on a single machine, while still offering the network enough leverage to form more complex, abstract representations to capture any hierarchical relations in the underlying structure of temporal dependencies. The resulting models are illustrated in figure 1 and 2, respectively.

IV. Experiments & Results

This section will cover the experiments executed to train the two models. First, a set of preliminary model checks will be briefly described. Second, the experiments performed to successfully train the base model will be reviewed. Third, the steps taken to train the extended model will be presented. Finally, the main results will be briefly summarized.

i. Setup

Prior to implementing our primary models, we execute a few preliminary experiments on the

base model (CNN) using only small subsets of the full training dataset.

Validation of implementation

First, we attempt to overfit our model to a very small data sample of only 20 random CT scans. Assuming a successful implementation of our data processing pipeline and model architecture, we expect to see fast convergence and high accuracy on the training set, but increasing loss on the validation set due to the small training sample size. The results do indeed indicate successful overfitting on the training set, confirming a valid implementation of the base model and data processing infrastructure.

Learning rate

Second, we attempt to identify an efficient learning rate, η, by running a sequence of five identical base models using the Adam optimizer with evenly spaced learning rates $\eta \in \{1e^{-1}, 1e^{-2}, \ldots, 1e^{-5}\}$. An efficient learning rate is characterized by a quick and smooth decrease towards a low and steady state of convergence in the loss function. We find that learning rates in the range of $\eta = 1e^{-1}$ to $\eta = 1e^{-2}$ do not lead steadily to convergence, but instead tend to cycle endlessly between vastly different loss values. This phenomenon of skipping from hill top to hill top suggests that the learning rates are too high and coarse for the given optimization problem. Conversely, learning rates in the range of $\eta = 1e^{-5}$ and below display steady decreases in the loss function, but are very slow to converge. Consequently, we consider learning rates in the interval from $\eta = 1e^{-3}$ to $\eta = 1e^{-4}$ efficient, demonstrating both fast and steady convergence towards a stable minimum in the loss function. We also test the Nesterov-Adam optimizer, characterized by automatically determining and adjusting the learning rate, but find it to be marginally less efficient when compared to the results of our previous experiments. In conclusion, therefore, learning rates of $\eta = 1e^{-3}$ are used in the following experiments unless explicitly stated otherwise.

Image downsampling

Third, we attempt to further reduce training time by experimenting with the size of input images fed to the network. As the CT scans are represented by large $142 \times 322 \times 262$ volumes, we may (most importantly) significantly reduce the amount of parameters needed in the fully connected layers, while simultaneously decreasing the number of convolution operations necessary in the convolutional layers of the network. A successful downsampling strategy of the input layers would thus lead to decreased training time while maintaining prediction performance. Since emphysema occurs in coherent patches of lung tissue, and since its extent is measured through visual inspection by experts, we downsample dynamically by selecting only specific voxels in coherent, non-overlapping $d \times d \times d$ blocks through pooling operations.

For instance, when downsampling by a factor of $d = 2$, we split the image into $2 \times 2 \times 2$ blocks, representing each block by the voxel containing the numerically highest value. In this way, by using max pooling over average pooling operations, we hope to retain as much of the original image texture as possible, as we know from visual inspection that emphysema is characterized by its unique textural pattern. While the results do indicate faster training, we unfortunately also find indications of dropping validation accuracy as the downsampling factor, d, increases (see model 1 in figure 3). This suggests that vital information is lost during compression of the input image, why we resort to the original, unscaled downsampling factor of $d = 1$ in the following.

Batch size

Finally, having fixated the size of our input, we vary the batch size in an attempt to minimize the total training time. By increasing the batch size, we minimize the expected variance of every parameter update as outliers become increasingly improbable. On the other hand, larger batch sizes may computationally cripple the data processing pipeline due to sheer memory constraints, causing increases in total

training time. Due to the lack of viable downsampling strategies for our input volumes and physical memory restrictions, the results indicate that batch sizes cannot exceed approximately 5 volumes per batch on a NVidia Titan X GPU. Anticipating the need for five concurrent input volumes per sample, corresponding to each time step in the extended model, we correspondingly restrict the batch size for our base model to a single input volume per batch. To further alleviate time constraints, we compensate by implementing parallel batch generation through the use of multiple, concurrent workers.

ii. Base model

Having identified and validated our basic configuration, we proceed to experiment with the implementation of our base model, GAPNet, using the full training dataset. The experiments will be presented in order of occurrence.

Sample size

First, we gradually increase the amount of training samples until the size of the full training set is reached. As the model contains a fixed amount of parameters, increasing the number of samples in our training set is expected to have a regularizing effect on the training session. In this way, the network is gradually forced into learning abstract, generalizable representations of the input data rather than simply dedicating its many parameters to memorizing an otherwise limited amount of input samples.

As expected, the results indicate an increasing delay in the onset of overfitting as the sample size increases for each training session. We do not, however, find any indications of the model learning generalizable representations as the validation loss remains approximately constant throughout the training session. Only during the first few epochs does the validation loss decrease, upon which it remains constant until the onset of overfitting, where it naturally starts to increase again (see model 2 in figure 3). Upon further examination of the predictions made by the resulting model, it is apparant

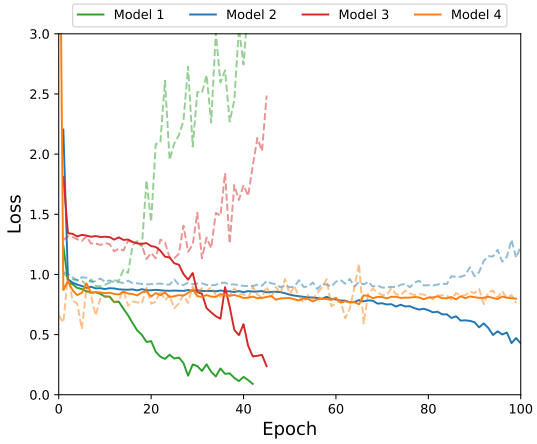

Figure 3: *Unsuccessful base model implementations. Solid and dashed lines indicate training and validation set performance, respectively.*

that the model quickly learns to recognize the class imbalance inherent in the dataset, and proceeds to exclusively predict the majority class following the first few epochs of training.

Class-weighted loss

Second, we therefore experiment with the use of class weights in the loss function. By calculating the relative weight of each class and incorporating them directly in the loss function, we may incur a stricter penalty to the model for wrongly predicting minority classes. In this way, we may effectively incentivize the model to further account for the minority classes if it is to decrease the total loss value. Even with a class weighted loss function, however, the model still appears to struggle in practice when discerning the minority classes, resorting instead to predicting only the majority class.

Oversampling

Third, then, we attempt to further increase the incentive to account for minority classes by oversampling them during training. Imitating the procedure used in Bortsova et al. (2018), we increase the batch size to three, reserving one entry for the majority class with no emphysema, another one for samples of light emphysema extent, and the last entry for samples of severe emphysema extent. As the latter two

minority classes naturally occur less frequently than the majority class samples, we randomly shuffle and repeat the minority class samples throughout the epoch until every majority class sample has been processed by the model exactly once.

This naturally increases the number of samples processed per epoch, since for every healthy sample in a batch, an additional mild and severe case will also be included in the batch (with repetition if necessary). Due to time constraints, we therefore resort to the reduced training set in an attempt to preserve the total training time. As indicated in table 1, this reduced dataset consists of only 100 randomly chosen images, thus respecting the overall distribution of the full dataset, except for the two most severe grades which are not represented at all. As a proof of concept, we choose not to simply draw a more balanced, reduced dataset, aiming instead to apply the current method to the full dataset in a future study. Since the remaining experiments will be drawing on oversampling, this reduced training set will also be utilized for the following experiments listed below. Evaluation, however, is still performed on the full validation set to ensure comparability of performance with previous models.

Although it is now increasingly expensive for the network to ignore the minority classes, both due to their increased cost in the loss function and their increased frequency of occurrence during training, in practice, the model nonetheless continues to predict only the majority class until the point of overfitting (see model 3 in figure 3). One possible explanation may be that, despite the minority class samples occurring more frequently, the model may still struggle to discern them. Further, by repeating the same minority classes during training, the model may erroneously be led to model emphysema as a less variable phenomenon than it really is. In effect, then, oversampling may be contributing more towards overfitting the training data than towards incentivizing the model to learn generalizable emphysema patterns.

Data augmentation

Fourth, we attempt to remedy this unintended effect by artificially increasing variability in the minority classes through preprocessing of the input data. Specifically, we randomly flip each input image among each of the three image dimensions. As the identifying characteristics of emphysema are not affected by being mirrored along a given axis, we can effectively increase the number of unique emphysema samples in our training set and decrease the amount of repetition during oversampling. Additionally, to further restrict the model's ability to overfit to the training data, we also randomly shift the input images up to 10 pixels in either direction along each of the three axes. Once again, however, the results show no indication of the model learning generalizable representations of the minority classes. With only a slight delay in the onset of overfitting, the model therefore retains its majority class predictions.

Regression & target rescaling

Fifth, we therefore alter the output layers in the model architecture to treat the input data as a regression problem instead of a classifcation task. More concretely, we now penalize an erroneously healthy prediction on a sample of severe emphysema more harshly than a healthy prediction on a light emphysema sample. In this way, the model may better utilize the metric information inherent in the target values and become further incentivized to account for severe cases of emphysema extent.

As the target values represent the proportion of lung tissue affected by emphysema, and the labelled intervals of estimated emphysema extent are not equally dispersed, it is relatively less costly to wrongly predict a case of low emphysema extent as healthy, since these target values are numerically more closely distributed than the target values representing cases of severe emphysema extent. As an additional experiment, we therefore rescale the target values to be evenly spaced, thus incentivizing the model to also distinguish between the individual minority classes. Again, however, in neither of the two experiments does the model deviate

from exclusive majority class predictions, why the validation loss remains relatively constant throughout the training session (see model 4 in figure 3).

Image cropping

Sixth, we scale down the complexity of the model by cropping the input images to fit the size of the smallest images in each independent dimension. In this way, larger images will have parts of the lung edge cropped off, but the overall dataset will have less background noise for the model to consider. Although emphysema may seem to only gather in local clusters, the condition is typically present in each region of the lung. Hence, despite possibly cropping off local emphysema clusters, cropping the images is expected to assist the model in focusing only on relevant parts of the image. The smaller input size simultaneously allows for the use of an increased batch size of 15 images. In combination with oversampling and rescaled regression target values, the model begins to learn after approximately 40 epochs of training with convergence occurring after around 270 epochs. With this model in hand, we proceed to implement the extended model. The results will be presented and evaluated in more detail in section V.

iii. Extended model

We now replicate the working model architecture for GAPNet for each of our five time steps, funnelling the resulting feature maps through to our extended model.

Naive approach

First, we replicate the successful GAPNet architecture across time steps adding only the recurrent extension module. In this approach, we naively seek to relearn the base model weights from scratch, allowing the base model freedom to learn slightly different features that might be more useful in the temporal case. In theory, this setup would permit the model to learn mutually dependent features across each individual time step. Given the sensitivity involved during training of the model in the single case, however, practical circumstances suggest a naive approach may not be feasible. The results support this intuition as the model does not even learn to identify the majority class, resorting entirely to random predictions.

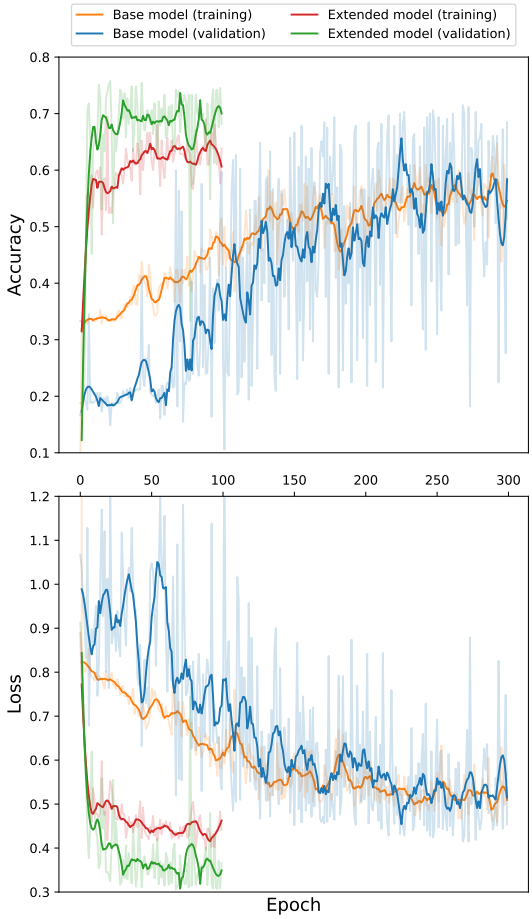

Figure 4: *Savitzky-Golay smoothened learning curves for the final base and extended models.*

Transfer learning with free base weights

Second, we import the weights of the successful base model achieving the lowest validation error to our extended model, but leave the network free to adjust the weights of both the preloaded base weights and the recurrent module during training. By using pretrained base weights, we hope to offer the network a headstart in identifying useful features for emphysema detection. Assuming these features

will not differ drastically in the temporal case, we expect the model to at most fine-tune the base weights to optimally learn the temporal dependencies in the recurrent module.

The results indicate a slight improvement over the naive model as the network no longer predicts at random by utilizing the pretrained weights. In spite of this, however, the model still manages to perform worse on the validation set than the base model itself. In other words, although the model does seem to be drawing on the prelearned features, it still manages to spoil this information in the temporal case. A tentative interpretation may be that the network is attempting to adapt the pretrained weights to extract features from one of the input images, but, since the weights are shared across time steps, accidentally ends up distorting the feature set for the other input images in the process.

Transfer learning with fixed base weights

Third, we again preload the extended model with the weights of the most successful base model, but restrict the model to only adjust the weights of the recurrent module. In this case, then, we force the model to only focus on modelling temporal dependencies by updating the weights in the recurrent module. Since emphysema is an irreversible condition, we may expect the model to identify a monotonically increasing relationship between the time steps, or, in the worst case, simply adopt an estimate based on the average predicted grade of each input image. Given that only the weights in the recurrent module are learnable, the model quickly converges after approximately 30 epochs. Contrary to the previous models, the results do indicate a slight improvement over the base model when evaluated on the validation set.

iv. Summary

For the given model and dataset, training proves to be a rather finicky affair. Reframing the task as a regression problem with equidistant targets, oversampling of minority classes, random shifting and flipping of the input, a reduced sample size of 100 cropped images, a batch size of 15, and using the Adam optimzer with a learning rate of $1e-3$ and a class weighted mean absolute error loss function, the base and extended models, in the latter case with fixed pretrained weights, do learn from the training data and successfully generalize to the validation set (see figure 4). In the next section, we will evaluate the base and extended models on the test set and discuss the results.

V. Evaluation

For both the base and extended models, we extract the weights offering the lowest validation set loss. Using these weights, test set emphysema grade predictions are then compared to the corresponding labels identified by expert observers. The resulting confusion matrices are illustrated in figure 5.

i. Base model

The base model offers an overall prediction accuracy of 69%, which, due to the inherent class imbalance in the dataset, is roughly equivalent to a hypothetical model predicting healthy subjects only. Comparing only predictions of healthy (0%) and non-healthy (1-100%) subjects, however, the model achieves an accuracy of 80%. This indicates that the model does capture more variance than a naive model, blindly predicting only the majority class, would, while also suggesting that the model struggles to accurately discern the exact grades of emphysema extent.

The confusion matrix shows that the best prediction accuracy is gained among predictions of healthy subjects, where only 11% of subjects are wrongly predicted as having a very light grade of emphysema (1-5%). Although the model does accurately capture between 22 and 30% of subjects with light to moderate emphysema extents (1-50%), the confusion matrix also seems to reveal a systematic prediction bias in this grade interval.

In particular, the base model seems to be consistently underestimating the true emphysema grade as determined by the expert observers.

For the lower grades, some confusion is in part to be expected, since the intervals are relatively narrow and closely aligned, but this does not explain the apparent, systematic bias among the moderate grade intervals. Although we cannot rule out the possibility of bias in expert observer labelling, an alternative explanation may arise from the underlying dataset closely resembling a Poisson distribution. Since lower grades are increasingly more common across classes, a similarly skewed distribution may also be found within classes, which may induce an incentive for the model to favour the lower ends of each grade interval bracket. While the use of oversampling during training should, to some extent, even out the effect across classes, the incentive may still persist if intra-class distributions are similarly skewed. If this is the case, however, we could expect the extended model to alleviate this issue by constraining the lower interval bound through the use of temporal dependencies.

Finally, for the severe cases (51-100%), the base model does seem to recognize some amount of emphysema, although the exact grade predictions range from very light to most severe. This is less surprising, however, given the lack of severe cases present in the reduced training set and the variance naturally bound to arise from the small amount of severe samples occurring in the test set. Again, by succesfully identifying temporal dependencies, the extended model may be able to utilize previous estimates to approach the true emphysema grade extents.

ii. Extended model

The extended model achieves an overall prediction score of 74%, indicating a slight improvement over the base model. The sensitivity in discerning healthy and unhealthy subjects is roughly untouched, however, with an accuracy of 81%, suggesting that the majority of the increase in overall prediction performance is attributable to improvements in emphysema grade distinction.

The confusion matrix supports this intuition as model predictions of healthy subjects only indicate slight improvements to a total accuracy of 91%. On the contrary, the extended model appears to perform slightly worse on very light grades of emphysema (1-5%), where

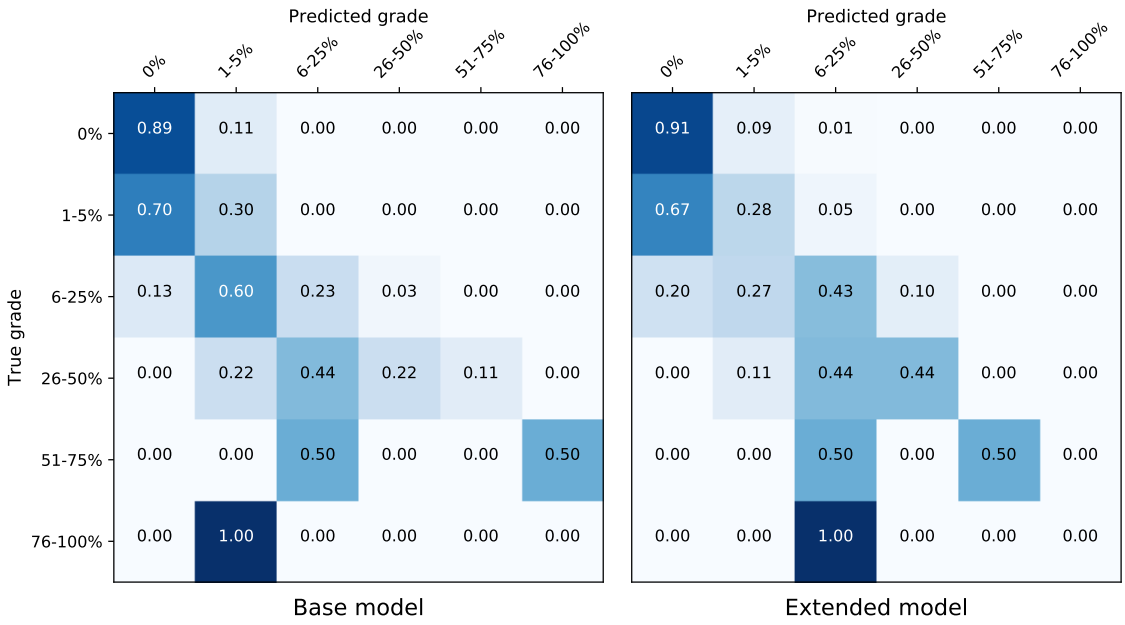

Figure 5: *Confusion matrices showing test set performance of the base model (left) and the extended model (right).*

only 28% of subjects are predicted in accordance with the expert observers' evaluations. As previously mentioned, however, this grade spans a very narrow interval, why some confusion is partially to be expected in this range.

Inspecting the light to moderate emphysema extents (6-50%), we find the main improvements over the base model. For these extents, accuracies of 43-44% are found, constituting a near-doubling of the accuracies offered in the base model. Although the extended model still displays a slight prediction bias and tends to underestimate moderate extents of emphysema, the majority weight of predictions are now shifted towards the true grades, as determined by the expert observers, with the exception of the 26-50% interval, where the model is still equally as likely to underestimate by a full grade interval.

In the severe cases (51-100%), the extended model appears to converge more towards the true grades, making less overall errors, than the base model. In this way, despite the lack of severe case samples in the training set, the extended model still manages to correctly classify one of only two cases in the 51-75% grade interval. This may suggest that the extended model does not merely resort to an average of the five input image predictions, but does in fact attempt to extrapolate from a monotonously increasing temporal correlation between them.

Given the short span of time over which the CT scans were collected, however, this may also be attributing too much credit to the model. Many potential sources of error arise in the collection of data, where many practical considerations, such as the proper inhalation of the test subject during scanning, may end up stirring the resulting images. For this reason, it is still possible that the extended model compensates for ambiguities in the final image by simply hedging its prediction estimates against previous images through a simple averaging strategy. As the base model consistently underestimates emphysema grades, however, and since, due to the irreverisbility of emphysema, previous images are unlikely to contain higher grades of emphysema extent, the extended model is more likely to utilize the information present in the additional input images to extract temporal dependencies in order to improve prediction performance over the base model.

VI. Conclusion

Convolutional neural networks offer a novel way to estimate the extent of emphysema in human lungs from CT scans. In this endeavour, we successfully demonstrate the capabilities of GAPNet, while simultaneously highlighting the finicky sensitivity involved in the training of such highly complex models. Through comprehensive use of input data augmentation, oversampling, target rescaling, and a class-weighted loss function, we achieve an overall accuracy of 69% on the test set with 80% of samples correctly distinguished as either healthy or unhealthy.

Extending the model with a recurrent neural network, we further show how historical, unlabelled data may be utilized to leverage prediction performance through the incorporation of temporal dependencies. By additionally training the recurrent module, an improved overall test set accuracy of 74% is achieved with 81% of subjects correctly predicted as healthy. Further, the extended model nearly doubles the prediction accuracy among light to moderate cases of emphysema, effectively correcting the underestimation bias present in the base model.

The study still leaves room for further investigation, however. In particular, since the full training set was abandoned due to time constraints, in favour of a reduced set, future studies may well achieve improved prediction performance by simply scaling the presented architecture and configuration to the full, uncropped dataset. Similarly, the extent to which temporal dependencies were succesfully captured by the recurrent module remains unclear, in part due to the scarcity of severe grade samples and, in part, due to natural limitations of variation in a slowly progressing disease captured by annual CT scans over the duration of only a five year period.

References

Ballas, N., Yao, L., Pal, C. and Courville, A. (2015), 'Delving deeper into convolutional networks for learning video representations', *arXiv preprint arXiv:1511.06432* .

Baumgartner, C. F., Kamnitsas, K., Matthew, J., Smith, S., Kainz, B. and Rueckert, D. (2016), Real-time standard scan plane detection and localisation in fetal ultrasound using fully convolutional neural networks, *in* 'International Conference on Medical Image Computing and Computer-Assisted Intervention', Springer, pp. 203–211.

Bortsova, G., Dubost, F., Ørting, S., Katramados, I., Hogeweg, L., Thomsen, L., Wille, M. and de Bruijne, M. (2018), Deep learning from label proportions for emphysema quantification, *in* 'Medical Image Computing and Computer Assisted Intervention (MICCAI)'. In.

Chen, H., Dou, Q., Ni, D., Cheng, J.-Z., Qin, J., Li, S. and Heng, P.-A. (2015), Automatic fetal ultrasound standard plane detection using knowledge transferred recurrent neural networks, *in* 'International Conference on Medical Image Computing and Computer-Assisted Intervention', Springer, pp. 507–514.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. and Darrell, T. (2015), Long-term recurrent convolutional networks for visual recognition and description, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2625–2634.

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural Comput* **9**(8), 1735–1780.

Ji, S., Xu, W., Yang, M. and Yu, K. (2013), '3d convolutional neural networks for human action recognition', *IEEE transactions on pattern analysis and machine intelligence* **35**(1), 221–231.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014), Large-scale video classification with convolutional neural networks, *in* 'Proceedings of the IEEE conference on Computer Vision and Pattern Recognition', pp. 1725–1732.

Kong, B., Zhan, Y., Shin, M., Denny, T. and Zhang, S. (2016), Recognizing end-diastole and end-systole frames via deep temporal regression network, *in* 'International conference on medical image computing and computer-assisted intervention', Springer, pp. 264–272.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B. and Sánchez, C. I. (2017), 'A survey on deep learning in medical image analysis', *Medical image analysis* **42**, 60–88.

Pedersen, J. H., Ashraf, H., Dirksen, A., Bach, K., Hansen, H., Toennesen, P., Thorsen, H., Brodersen, J., Skov, B. G., Døssing, M. et al. (2009), 'The danish randomized lung cancer ct screening trial—overall design and results of the prevalence round', *Journal of Thoracic Oncology* **4**(5), 608–614.

Poudel, R. P., Lamata, P. and Montana, G. (2016), Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation, *in* 'Reconstruction, Segmentation, and Analysis of Medical Images', Springer, pp. 83–94.

Simonyan, K. and Zisserman, A. (2014), Two-stream convolutional networks for action recognition in videos, *in* 'Advances in neural information processing systems', pp. 568–576.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M. (2015), Learning spatiotemporal features with 3d convolutional networks, *in* 'Computer Vision (ICCV), 2015 IEEE International Conference on', IEEE, pp. 4489–4497.

Wille, M. M. W., Thomsen, L. H., Dirksen, A., Petersen, J., Pedersen, J. H. and Shaker, S. B. (2014), 'Emphysema progression is visually detectable in low-dose ct in continuous but not in former smokers', *European radiology* **24**(11), 2692–2699.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H. and Courville, A. (2015), Describing videos by exploiting temporal structure, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 4507–4515.

Ypsilantis, P.-P. and Montana, G. (2016), 'Recurrent convolutional networks for pulmonary nodule detection in ct imaging', *arXiv preprint arXiv:1609.09143* .