Medical Image Analysis 55 (2019) 15–26

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Pulmonary nodule detection in CT scans with equivariant CNNs

Marysia Winkels[a,b,*], Taco S. Cohen[a]

[a] *University of Amsterdam, Netherlands*
[b] *Aidence B.V., Netherlands*

ARTICLE INFO

Article history:
Received 15 August 2018
Revised 22 March 2019
Accepted 26 March 2019
Available online 28 March 2019

Keywords:
Deep learning
Convolutional neural networks
Equivariance
Pulmonary nodule detection
CT scans

ABSTRACT

Convolutional Neural Networks (CNNs) require a large amount of annotated data to learn from, which is often difficult to obtain for medical imaging problems. In this work we show that the sample complexity of CNNs can be significantly improved by using 3D roto-translation group convolutions instead of standard translational convolutions. 3D CNNs with group convolutions (3D G-CNNs) were applied to the problem of false positive reduction for pulmonary nodule detection in CT scans, and proved to be substantially more effective in terms of accuracy, sensitivity to malignant nodules, and speed of convergence compared to a strong and comparable baseline architecture with regular convolutions, extensive data augmentation and a similar number of parameters. For every dataset size tested, the G-CNN achieved a FROC score close to the CNN trained on ten times more data.

© 2019 Published by Elsevier B.V.

1. Introduction

Lung cancer is currently the leading cause of cancer-related death worldwide, accounting for an estimated 1.7 million deaths globally each year (Wang et al., 2016) – a death toll larger than breast cancer, colon cancer and prostate cancer combined (American Cancer Society Statistics Center, 2017). This high mortality rate can largely be attributed to the fact that the majority of lung cancer cases is diagnosed when the cancer has already metastasised, as symptoms generally do not present themselves until the cancer is at a late stage (American Cancer Society, 2017).

Screening of high risk groups, where a reading radiologist would be tasked with identifying suspect lesions in the form of pulmonary nodules on chest CT images, could potentially increase early detection and thereby improve the survival rate (National Lung Screening Trial, 2011; Oudkerk et al., 2017). Many factors contribute to the effectiveness of screening, amongst which are the skill, alertness and experience level of the reading radiologist, as potentially malignant lesions are easy to overlook due to the rich vascular structure of the lung (see Fig. 1).

One way to reduce such observational oversights would be to use double readings (Lauritzen et al., 2016; Wormanns et al., 2005), a practice in which two readers independently interpret an image and combine findings, but this would also drastically add to the already increasing workload of the radiologist (Bhargavan et al., 2009), and increase the cost of care. A much more cost-effective approach would be to introduce computer aided detection (CAD) software as a second reader to assist in the detection of lung nodules (Bogoni et al., 2012; Zhao et al., 2012).

1.1. Challenge

For automated medical image analysis, deep learning and in particular the Convolutional Neural Network (CNN) has become the methodology of choice. With regards to pulmonary nodule detection specifically, deep learning techniques unambiguously outperform classical machine learning approaches (van Ginneken, 2017; Firmino et al., 2014; Al Mohammad et al., 2017), making CAD a potentially valuable addition to the radiologists' toolbox. However, CNNs typically require a substantial amount of labeled data to train on – something that is hard to come by, both due to patient privacy concerns and the labor-intensity of obtaining high-quality annotations. The problem is further compounded by the fact that in all likelihood, the data collection process will have to be repeated for different imaging modalities, scanner types, scanner settings, resolutions, image reconstruction techniques and patient populations, because current methods often fail to generalize across such variability if it is not present in the training data.

The challenge this presents is that of *data efficiency*: the ability to learn in complex domains without requiring large quantities of data. This is the challenge we address in this paper, using the problem of pulmonary nodule detection in CT scans as an important test case of a more general technique.

[*] Corresponding author at: University of Amsterdam, Netherlands.
E-mail addresses: marysia@aidence.com, marysia@live.nl (M. Winkels).

https://doi.org/10.1016/j.media.2019.03.010
1361-8415/© 2019 Published by Elsevier B.V.

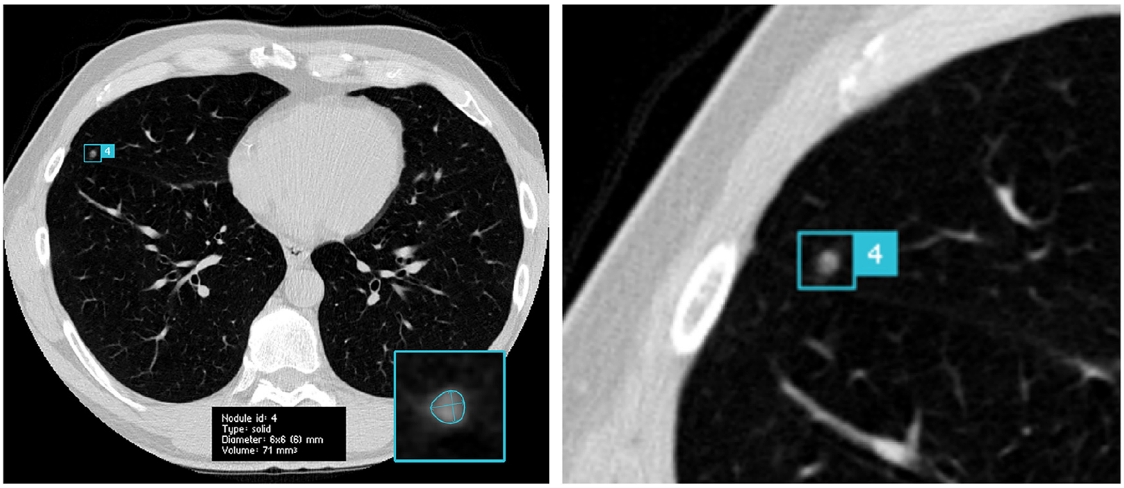


Fig. 1. Lung nodule on axial thorax CT.

1.2. Approach

Our strategy for improving data efficiency is to exploit prior knowledge about the symmetries of the problem. Specifically, for most medical imaging problems, and certainly for pulmonary nodule detection, we know a priori that the relevant patterns can occur not just at any position in the image, but also in any orientation. Whereas CNNs are good at exploiting translation symmetries, they do not have a built-in ability to generalize across orientations and reflections.

In order to achieve this, we develop 3D *G*-CNNs, a 3D version of group equivariant convolutional networks presented in Cohen and Welling (2016), that generalizes automatically over discrete rotations and reflections. 3D G-CNNs extend convolutional weight sharing from translations to rotations and reflections by using group convolutions, and are carefully designed to be *equivariant*: a transformation of the input of a layer leads to a transformation of the output of that layer. Because of this property, each layer of the *G*-CNN preserves the symmetry, allowing further layers to exploit it via weight sharing.

1.3. Contribution

We apply 3D G-CNNs to the task of pulmonary nodule classification – the last step in the nodule detection pipeline for chest CT scans that aims to reduce the number of false positive reports by a CAD system (Firmino et al., 2014; Al Mohammad et al., 2017). G-CNNs show remarkable data efficiency, yielding similar performance to CNNs trained on 10 × more data (with data augmentation), without any additional tuning. Beyond data efficiency, we show that G-CNNs require fewer updates to converge. Finally, we observed that on this dataset, G-CNNs seem to be more sensitive to malignant nodules.

Our implementation of 3D group convolutions is publicly available[1], so that using them is as easy as replacing `Conv3D()` by `GConv3D()`.

1.4. Outline

In what follows, we will first provide an intuitive notion as to why conventional convolutional layers could be generalised to reduce the sample complexity, briefly touched upon in Section 1.2, and provide an overview into related work on this subject (Section 2). Next, Section 3 will describe group convolutions for the specific case of 3D rotations and reflections, and elaborate on how this ensures equivariance. Section 4 provides a description of the experimental setup, including datasets, evaluation protocol network architectures, and Section 5 compares G-CNNs to conventional CNNs in terms of performance under various conditions and studies the rate of convergence. We will discuss these results and conclude in Sections 6 and 7, respectively.

2. Background & related work

Convolutional neural networks are already quite data efficient compared to fully-connected neural networks. This is due to translational *weight sharing* across the input image, which leads to translational *equivariance* – because the weights are shared across the input, a shift in the input leads to a corresponding shift in the output.

In turn, the translation equivariance of the individual convolutional layers, combined with the dimensionality reduction that removes spatial information, make convolutional neural networks *invariant* to translation: a translational shift in the input image generally does not influence the ability of a trained model to correctly identify its contents.

Translational invariance contributes to the effectiveness of CNNs in the domain of image analysis as images exhibit translational symmetry; local visual patterns can appear in every position with equal probability, and should be classified in the same way regardless of position. However, many kinds of patterns – including pulmonary nodules – not only maintain their identity under translation, but also under other transformations such as rotation and reflection. It is therefore natural to ask if the translational weight sharing of convolutional neural networks can be generalised to other kinds of transformations by extending the equivariant property of the convolutional layer to other transformations.

One common and effective way to obtain approximate invariance towards a set of transformations is to simply augment the existing training dataset with said transformations (Simard et al., 2003). Although this generally does improve performance, it does not guarantee invariance of the network on held out test data or even on the training data. Moreover, data augmentation puts a soft constraint on the network as a whole, whereas we constrain *every layer* of the network to be exactly equivariant. Layerwise equivariance is a significantly stronger constraint than invariance, because it requires the network to not only output the same thing

[1] https://github.com/tscohen/GrouPy.

for all transformed versions of an input, but also to process the input image *in the same way*, no matter how it is transformed. More specifically, each feature map in a G-CNN will extract exactly the same features and contain the same information about the input, regardless of any transformation applied to the input (as long as this transformation is in the group G under consideration).

Although it would in principle be nice to *learn* equivariance, as a practical matter and given the current state of technology, it seems that building equivariance into the network is more effective. Indeed, the popularity of translational CNNs shows that replacing fully connected neural networks (with no built-in equivariance) with CNNs is very beneficial. In this paper we show that including rotations and reflections gives a similarly significant boost in performance.

Several authors have investigated rotation equivariant networks. Dieleman et al. (2016) introduced operations that can be directly inserted into an existing network architecture that exploit cyclic symmetry by maintaining multiple (rotated) feature maps at every layer of the network. They specifically note that creating rotated feature maps is equivalent to applying rotated filters on the original input, the advantage of the former being ease of implementation, but at the cost of increased memory requirements. Applying a rotated transformation on the filters as we do in this paper is more memory efficient, because filters are generally smaller (*e.g.* 3×3) than feature maps.

Whereas Dieleman et al. are more focused on efficient implementation for the specific case of 90^o rotations, Cohen and Welling (2016) provide a general framework based on symmetry groups that can be easily extended. In their work, they introduce *group convolution* layers for discrete groups based on filter transformation that can be added as a drop-in replacement for spatial convolutions in CNN architectures, with negligible computational overhead.

There is now a growing number of papers on equivariant convolutional networks, with G-CNNs being developed for discrete 2D rotation and reflection symmetries (Cohen and Welling, 2016; Dieleman et al., 2016; Cohen and Welling, 2017), continuous planar rotations (Worrall et al., 2017; Weiler et al., 2018b, 2018a; Bekkers et al., 2018), 3D rotations of spherical signals (Cohen et al., 2018), and permutations of nodes in a graph (Kondor et al., 2018). In this paper, we develop G-CNNs for three-dimensional signals such as volumetric CT images, acted on by discrete translations, rotations, and reflections. This is highly non-trivial, because the discrete roto-reflection groups in three dimensions are non-commutative and have a highly intricate structure (see Fig. 4).

3. 3D group convolutions

In this section, we will introduce group convolutions for the specific case of 3D rotations and reflections, without using abstract concepts from group theory. In contrast to earlier expositions of G-CNNs, here we will stay fairly close to the implementation level. For the theoretical framework, we refer the reader to Cohen and Welling (2016).

The key observation at the basis of group convolutions is this: instead of exclusively translating a filter over the input (as one would do in a conventional convolution), other types of transformations can be applied to the filter as well, thereby increasing the degree of weight sharing. The transformations we consider in this paper are discrete 3D rotations and reflections. We focus on discrete (*e.g.* 90°) rotations only for the time being, because these can be applied losslessly to a filter without interpolation, and these already lead to a substantial improved performance. We will detail the considered set of transformations in Section 3.1.

To summarize, in addition to applying the same filters solely at every *position* in the input, we also apply them in every *orientation*. This point of view suggests a simple and efficient method of implementation, where to compute the group convolution, each transformation (a rotation and/or reflection) is applied to each filter, and the input is then convolved with all these original and newly created transformed filters in a single call to `Conv3d`.

However, whereas a conventional CNN produces one output channel per filter, a G-CNN produces several *orientation channels* per filter, one for each transformation that was applied. The feature spaces, which have multiple orientation channels, transform differently from the input space, as the orientation channels (corresponding to a given filter) get *shuffled* in addition to being transformed when the input is transformed.

This means that we need to consider two situations: the first layer of the network, where a group convolution is applied to a 3D volume; and the subsequent layers, where the group convolution is applied to the output feature maps of a group convolution (with orientation channels) and the channel shuffling phenomenon occurs. We will discuss these situations (and how equivariance is ensured) in Sections 3.2 and 3.3, respectively.

3.1. Rotations and reflections in 3D

We are specifically interested in symmetries of the *label function*: transformations that, when applied to the input, do not alter the predicted label. For most medical imaging problems, and for lung nodule detection in particular, rotations and reflections are symmetries – a nodule must be classified as such, regardless of the orientation in the lungs. From this follows the idea that if a given filter is useful for a task, rotated copies of this filter may be equally useful.

However, applying an arbitrary continuous rotation (*e.g.* 45°) to a small (e.g. $3 \times 3 \times 3$) filter is tricky, because it requires padding and interpolation, which can introduce computational overhead and numerical artefacts. Moreover, a fine discretization of the space of continuous rotations would result in a large number of filter transformations, and hence a large number of channels, making it computationally impractical[2] Although both of these issues may be solvable (see for instance recent papers by Worrall et al. (2017), Weiler et al. (2018b), Bekkers et al. (2018), Thomas et al. (2018) and Weiler et al. (2018a), in this paper we have opted for the simplest, most straightforward and most robust approach, which is to work with discrete rotations and reflections, which can be applied easily to small filters without padding and interpolation.

This still leaves us with some choices, first of which is the set of transformations to apply to the filter. While a regular filter of size $S \times S \times S$ resembles a cube, for many different kinds of 3D imaging modalities such as CT and MRI, the z-spacing of voxels is different from the x- and y-spacing, which means the voxels themselves (and therefore the $S \times S \times S$ filter that applies to that input) corresponds to a rectangular cuboid with a square base. Whereas a cube (Fig. 2 (left)) has 24 orientation-preserving symmetries (the *octahedral group* O), the rectangular cuboid (Fig. 2 (right)) only has 8 (the group D_4).

Because of the difference in spacing, one could hypothesize that a pattern in upright position should not be rotated to a horizontal position, because that would result in a Z-axis squashing of the pattern. On the other hand, such a squashed pattern may still be useful, so ultimately the choice between the O and D_4 groups remains an empirical question, which will be studied in this paper.

In addition to the choice between cube and cuboid symmetry, there is the choice of whether to consider only orientation-preserving symmetries (rotations) or reflections as well. Thus, we

[2] Steerable CNNs (Cohen and Welling, 2017) provide a solution to this issue, by working with low-dimensional representations of large symmetry groups, but are beyond the scope of the present paper.

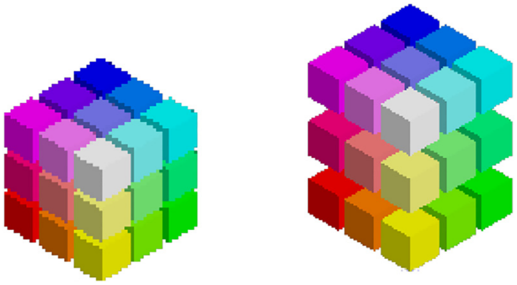


Fig. 2. A $3 \times 3 \times 3$ filter could correspond to a cube (left) or rectangular cuboid with square base (right) in physical space, depending on the Z-axis spacing of the scanner.

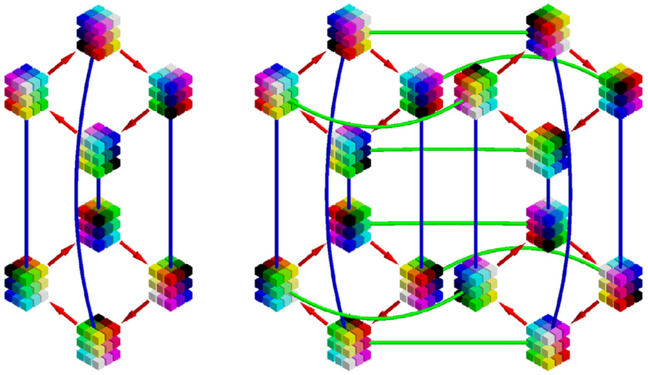


Fig. 3. Cayley diagrams of the groups D_4 (left) and D_{4h} (right). Red arrows correspond to 90° rotation around the Z-axis, blue lines correspond to 180° rotations around the X-axis, and green lines correspond to reflections in the XY-plane. Best viewed in color. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

end up with four symmetry groups of interest: the orientation-preserving and non-orientation preserving symmetries of a rectangular cuboid (D_4 and D_{4h}, respectively), and the orientation-preserving and non-orientation-preserving symmetries of a cube (O and O_h, respectively). We will refer to these groups generically using the letter H. The choice of symmetry group will be studied empirically in Section 5, and discussed in Section 6.

In order to discuss group convolutions, and the channel shuffling phenomenon alluded to in Section 3, we aim to create an understanding of how the transformations in these four groups work in a visual manner. Figs. 3 and 4, show the *Cayley diagram* for the groups D_4, D_{4h} and O (the group O_h is not shown, because with 48 elements, it is too large to easily visualize). In a Cayley diagram, each node corresponds to a symmetry transformation $h \in H$, which is visualized in these figures by its effect on a canonical $3 \times 3 \times 3$ filter. The nodes in the diagram are connected by lines and arrows of various colours. Following the coloured line illustrates what happens if you apply a particular *generator* transformation to the associated canonical filter. Applying the generators in sequence (*i.e.* visually following the lines and/or arrows in the diagram), any transformation in the group can be made. As different sequences of generator transformations (paths) can lead to the same node, there can be multiple distinct paths between two nodes, leading to an intricate graph structure.

Example. Fig. 3(left) shows the Cayley diagram of the group D_4 (orientation-preserving symmetries of a rectangular cuboid). The generators of this group are the 90° rotation around the Z-axis (red arrow) and the 180° rotation around the Y-axis (blue line). The latter is shown as a line instead of an arrow, because it is self inverse: $h^{-1} = h$. We see that this group is not commutative, because starting from any node, following a red arrow and then a blue line leaves us in a different place from following a blue line and then a red arrow. Similarly, Fig. 3 (right) shows the Cayley diagram for D_{4h} (non-orientation-preserving symmetries of a rectangular cuboid). This diagram has an additional generator, the reflection in the Z-plane (drawn as a green line). Fig. 4 shows the Cayley diagram for O, the orientation-preserving symmetries of a cubic filter, generated by Z-axis rotations (red arrows) and rotations around a diagonal axis (blue arrows).

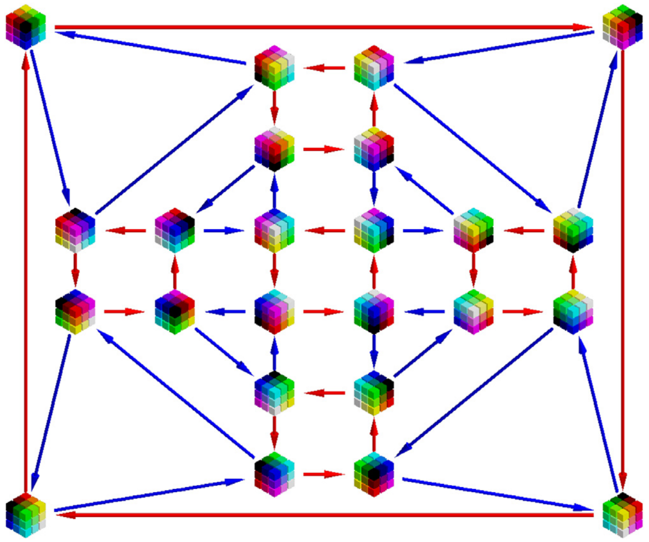


Fig. 4. Cayley diagram for O. Red arrows correspond to Z-axis rotation, whereas blue arrows correspond to rotation around a diagonal axis. Best viewed in color. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Anticipating our discussion of channel shuffling, it is useful to observe that each generator h (arrow or line color) defines a *permutation* of the elements of the group. This permutation sends each element h' to the element hh', which corresponds to the graph node connected to h' by the outgoing arrow associated with h. For instance, the red arrow in Fig. 3 (left) corresponding to a Z-axis rotation by 90 degrees defines a simultaneous cyclic shift among the four elements drawn on top, and those drawn at the bottom. Likewise, the permutation corresponding to the blue line exchanges the top and bottom components of the figure.

Since each element of the group can be written as a sequence of generators, and each generator corresponds to a permutation, we can easily infer the permutation corresponding to each element of the group by multiplying generator permutations. The identity transformation (every group has an identity element) is assigned the identity permutation. In a G-CNN, each feature has one orientation channel for each element of H, and as we will see, these channels get permuted in the way just described, whenever the input is rotated or reflected.

3.2. Group convolution for 3D images

The first step of the group convolution is to apply every transformation h in the chosen group H to each of the filters. To describe this more precisely, we introduce the following notation. Let Ψ denote the filter bank, stored as an array of size $C \times K \times S \times S \times S$, where C is the number of output channels, K the number of input channels, and S the spatial size of the filter. We say that Ψ consists of C filters, each having K channels[3] If we denote each filter

[3] Sometimes, each of the $C \cdot K$ blocks of size $S \times S \times S$ is called a filter, but we refer to this as a filter *channel*.

channel by ψ_c^k (for $c = 1, \ldots, C$ and $k = 1, \ldots, K$), we can write the filter bank as a matrix:

$$\Psi = \begin{bmatrix} \psi_1^1 & \cdots & \psi_1^K \\ \vdots & \ddots & \vdots \\ \psi_C^1 & \cdots & \psi_C^K \end{bmatrix} \tag{1}$$

Note that each element of this matrix represents a 3D subarray of size $S \times S \times S$ (the filter channel). To be clear, we would like to emphasize that the notation introduced so far is not specific to G-CNNs, but applies to normal CNNs as well.

The transformation (roto-reflection) of an individual filter channel ψ_c^k by $h \in H$ is denoted $T_h \psi_c^k$. We will use an arbitrary but fixed ordering on the elements of H. Thus we can write $H = \{h_i\}_{i=1}^N$ where $N = |H|$ (the number of transformations in the group). With this notation in place, we can define the transformation of the filter bank as follows:

$$\Psi^+ = \begin{bmatrix} T_{h_1}\psi_1^1 & \cdots & T_{h_1}\psi_1^K \\ \vdots & \ddots & \vdots \\ T_{h_N}\psi_1^1 & \cdots & T_{h_N}\psi_1^K \\ \vdots & & \vdots \\ T_{h_1}\psi_C^1 & \cdots & T_{h_1}\psi_C^K \\ \vdots & \ddots & \vdots \\ T_{h_N}\psi_C^1 & \cdots & T_{h_N}\psi_C^K \end{bmatrix} \tag{2}$$

Note that for each filter $\psi_c = (\psi_c^1, \ldots, \psi_c^K)$ in Ψ (a row in Eq. (1)), there are $N = |H|$ filters of the form $(T_{h_i}\psi_c^1, \ldots, T_{h_i}\psi_c^K)$ (for $i = 1, \ldots, N$) in Ψ^+, stacked along the output channel axis.

If we use this augmented filter bank Ψ^+ in a `Conv3d`, we will produce $C \cdot N$ output channels. If we had used Ψ, we would get C channels only, even though it has the same number of parameters as Ψ^+.

The output feature maps are indexed by a feature index $c = 1, \ldots, C$ and a transformation index $i = 1, \ldots, N$. We will refer to channel (c, i) as the ith orientation channel of feature c.

The group convolution is implemented as

$$\texttt{GConv3d}(\Psi, f) = \texttt{Conv3d}(\Psi^+, f). \tag{3}$$

The map $\Psi \mapsto \Psi^+$ can be defined as an indexing operation of Ψ using a precomputed array of indices. This map is clearly differentiable, and so we can easily backpropagate gradients to the parameters Ψ in order to learn them by gradient descent.

3.2.1. Equivariance

What happens to the feature maps if we transform the input? This is important to understand, because only if we know the transformation behaviour of the feature space, can we perform the appropriate kind of weight sharing in the next layer.

In order to answer this question, notice first that the result of cross-correlating[4] a rotated input $T_h f$ with a rotated filter $T_h \psi$ is the same as the result of cross-correlating f with ψ directly, except that it is rotated:

$$(T_h \psi_c) \star (T_h f) = T_h(\psi_c \star f) \tag{4}$$

Using Eq. (4) as a rule of calculation, we find that we can express the correlation of ψ_c with the rotated input $T_h f$ in terms of the correlation of the original (non-rotated) input f with a back-rotated filter:

$$\psi_c \star (T_h f) = (T_h T_{h^{-1}} \psi_c) \star (T_h f) = T_h((T_{h^{-1}} \psi_c) \star f). \tag{5}$$

Consider now what this means when we apply the transformed filter bank Ψ^+ to a rotated input $T_h f$. Since this filter bank contains *all* the transformed copies of a given filter ψ_c, the back-rotated filter $T_{h^{-1}}\psi_c$ that we encountered in Eq. (5) is already in Ψ^+. So the right-hand side of Eq. (5) tells us that if we rotate the input by h, the response in orientation channel i of feature c will undergo a rotation by h, *and* move to a different orientation channel i'.

How exactly do the channels get shuffled? If we apply Eq. (5) to the ith orientation of filter c, i.e. to $T_{h_i}\psi_c$, and choose $h = h_j$ we find:

$$\left(T_{h_i}\psi_c\right) \star \left(T_{h_j} f\right) = T_{h_j}\left(\left(T_{h_j^{-1}} T_{h_i} \psi_c\right) \star f\right). \tag{6}$$

Now, since $T_{h_j^{-1}} T_{h_i}$ is just $T_{h_j^{-1} h_i}$, we see that the information in the ith orientation channel moves to the channel $i' = \iota(h_j^{-1} h_i)$, where ι (defined by $\iota(h_i) = i$) returns the index of a transformation in H.

We can summarize all of this in a matrix of indices $\iota_{ij} = \iota(h_j^{-1} h_i)$, which can easily be precomputed. The indices in this matrix can also be inferred from the Cayley diagram, *e.g.* Fig. 4. To do so, first fix a choice of index for each vertex in the diagram. Remember that each transformation h_j can be written as a sequence of generator transformations, corresponding to a sequence of red or blue arrows in the diagram. Then ι_{ij} is the index of the vertex one ends up in, if one starts at vertex i, and follows the path corresponding to h_j in reverse direction.

3.3. Group convolution for 3D images with orientation channels

The group convolution introduced in Section 3.2 takes as input an image without orientation channels, but produces feature maps with orientation channels. This is appropriate for the first layer of the network, but since the orientation channels get shuffled when the input is rotated, the group convolution used in the second and higher layers should take this into account. The basic idea remains the same, though: we apply each transformation $h \in H$ to each filter, only now the orientation channels of the filter also get shuffled when h is applied.

We can again write down the filter bank Ψ (shape $C \times K \cdot N \times S \times S \times S$), whose input channels now come in groups of N orientation channels:

$$\Psi = \begin{bmatrix} \psi_1^{1,1} & \cdots & \psi_1^{1,N} & \cdots & \psi_1^{K,1} & \cdots & \psi_1^{K,N} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \psi_C^1 & \cdots & \psi_C^{1,N} & \cdots & \psi_C^{K,1} & \cdots & \psi_C^{K,N} \end{bmatrix} \tag{7}$$

Here $\psi_c^{k,i}$ denotes the ith orientation channel of input feature k, for filter c. We want to emphasize that the weights corresponding to different input orientation channels are not tied in any way. We can store Ψ as a standard filter bank with $K \cdot N$ input channels, by collapsing the orientation channel and input feature indices i and k into a single input channel index.

Consider now a set of orientation channels $\Psi_c^k = (\psi_c^{k,1}, \ldots, \psi_c^{k,N})$ corresponding to a filter c and input feature k. To rotate Ψ_c^k, we apply a rotation to each orientation channel, and permute them with using ι,

$$\Psi_c^{k+} = \begin{bmatrix} T_{h_1}\psi_c^{k,\iota_{11}} & \ldots & T_{h_1}\psi_c^{k,\iota_{1N}} \\ \vdots & & \vdots \\ T_{h_N}\psi_c^{k,\iota_{N1}} & \ldots & T_{h_N}\psi_c^{k,\iota_{NN}} \end{bmatrix} \tag{8}$$

Conceptually, the filter bank Ψ^+ is then built by stacking these blocks Ψ_c^{k+} along the C and K axes. In practice, the map $\Psi \mapsto \Psi^+$ can again be implemented as a single indexing operation. As before, the group convolution can then be implemented as $\texttt{GConv3d}(\Psi, f) = \texttt{Conv3d}(\Psi^+, f)$.

[4] Despite the name, convolutional networks typically use cross-correlation instead of convolution in the forward pass.

3.3.1. *Equivariance*

This group convolution with channel shuffling is also equivariant, meaning that if the input is transformed (with channel shuffling), the output transforms in the same way. To see this, note that Eqs. (4)–(6) still hold if we redefine T_h to perform not just spatial rotation but also channel shuffling. Hence, the argument goes through unchanged.

3.4. *Group convolutional neural networks*

The two kinds of group convolutions (where the inputs contain orientation channels or not, described in Sections 3.2 and 3.3, respectively), we can create a deep network that is equivariant as a whole. In the first layer, one uses a group convolution for inputs without orientation channels. This layer produces an output with orientation channels, so from the second layer onwards one uses the second kind of group convolution. In between convolutions, one can use any kind of non-linearity, as well as (equivariant) batch normalization, skip connections, and other tricks (Cohen and Welling, 2016).

When creating a G-CNN starting from a conventional CNN baseline, it is important to note that the number of 3D channels will increase by a factor of $N = |H|$ unless one reduces the number of channels C. This increases computational cost, but more importantly, increases the number of parameters required per filter in the next layer. To avoid this issue and obtain a network with roughly the same number of parameters as the baseline network, one can divide the number of channels in each layer by $\sqrt{|H|}$. The new network will have $C/\sqrt{|H|}$ G-channels, each of which has $|H|$ orientation channels, giving a total of $|H|C/\sqrt{|H|} = \sqrt{H}C$ 3D-channels.Thus, the new network has $\sqrt{|H|}$ times *more* 3D feature maps, and thus $\sqrt{|H|}$ times more parameters *per filter* than the CNN baseline with C channels. Since we also reduce the number of channels (i.e. filters) in the next layers by $\sqrt{|H|}$, the number of parameters in the second and higher layers is the same as for the baseline.

However, as the number of 3D-channels ($|H|C/\sqrt{|H|}$, or $\sqrt{H}C$) is higher than in the original network (C), this new network will lead to an increase in computational cost. We can, instead choose to enforce an equal compute by dividing C by $|H|$ instead, as $|H|C/|H| = C$.

Another thing to keep in mind when using G-CNNs that, although some properties and features may exhibit rotational and reflectional symmetry, others may not. For example, one could argue that although pulmonary nodules themselves are symmetrical with regards to rotations and reflection, the surrounding structures (which could aid in determining the nature of the nodule) may not be. This issue can be easily circumvented by using a fully-connected layer as the last layer. This layer can learn that certain orientations of features should be favoured over others, thereby capturing that certain aspects (such as the surrounding structure) may not be symmetrical, while the equivariant base network can efficiently represent the input in a direction-agnostic manner.

4. Experimental setup

Modern pulmonary nodule detection systems typically consist of the following five subsystems: data acquisition (obtaining the medical images), preprocessing (to improve image quality), segmentation (to separate lung tissue from other organs and tissues on the chest CT), localisation (detecting suspect lesions and potential nodule candidates) and false positive reduction (classification of found nodule candidates as nodule or non-nodule) (Firmino et al., 2014; Al Mohammad et al., 2017). All the experiments in this paper are performed on a fixed dataset of *nodule candidates* produced by a state of the art pipeline. This reduces the false positive reduction problem to a relatively straightforward classification problem, and thus enables a clean comparison between CNNs and G-CNNs, evaluated under identical circumstances.

To determine whether a G-CNN is indeed beneficial for false positive reduction, we compare the performance of networks with group convolutions for various 3D groups $H = O, O_h, D_4$ or D_{4h}, (see Section 3.1) to a strong baseline network with regular 3D convolutions. In order to assess improvements in data efficiency, we repeat this experiment for datasets of size 30, 300, 3000 and $30,000$ samples. By evaluating on different dataset sizes, we are able to compare the performance gains due to group convolutions with those that can be obtained by expanding the dataset.

Data augmentation is a standard technique used to improve generalization across symmetry transformations. It is thus important to determine whether or not G-CNNs provide any benefit when data augmentation is used. For this reason, we use a state of the art data augmentation pipeline in all experiments (both CNN and G-CNN). This pipeline consists of random continuous rotation by $0 - 360^o$, reflection over all axes, small translations over all axes, scaling between $.8 - 1.2$, added noise, and value remapping. The general finding is that even in this context, G-CNNs provide significant benefits.

Another prevalent technique for exploiting prior knowledge of symmetries is to perform *test-time augmentation*. In this approach, the predictions made by a non-equivariant network for several transformed inputs are averaged to obtain an invariant prediction. We compare G-CNNs without test-time augmentation to CNNs with test-time augmentation (Section 5.2), and find that G-CNNs still outperform.

The reason for comparing G-CNNs with different symmetry groups is twofold. Firstly, we would like to know if the difference in pixel spacing in the Z direction relative to the X and Y direction means that we should use the cuboid groups (D_4 or D_{4h}), or whether we can just as well use the full cube symmetry groups (O or O_h), even when the filters do not represent a cubic volume.

Secondly, we wish to investigate the effect of the size of the symmetry group on generalization performance. On the one hand, one could hypothesise that since larger groups result in more weight sharing, larger groups are always preferable. However, a larger group also means a larger number of orientation channels, which increases the number of parameters per filter (because a filter in layer $l > 1$ has $N = |H|$ input orientation channels per input feature). Thus, at a fixed parameter budget, using a larger group means that fewer truly distinct patterns can be detected (but they can be detected in more orientations).

Additional experiments were performed specifically related to malignancy and speed of convergence, the details of which will be expanded upon in Section 5.

4.1. *Datasets*

The scans used for the experiments originate from the NLST (National Lung Screening Trial, 2011) and LIDC / IDRI (McNitt-Gray et al., 2007) datasets. The NLST dataset contains scans from the CT arm of the National Lung Screening Trial, a randomized controlled trial to determine whether screening with low-dose CT (without contrast) reduces the mortality from lung cancer in high-risk individuals relative to screening with chest radiography.[5]

The LIDC/IDRI dataset was created by the Lung Image Database Consortium (LIDC) in collaboration wit the Image Database Resource Initiative (IDRI) with the purpose of stimulating research

[5] The NLST dataset provides slice numbers for each annotation. The *xy*-coordinates were provided by Aidence' in-house radiologists.

Table 1
Specifics of the training, validation and test set sizes and class ratios.

Set	Source	Candidates	Positive %	Negative %
Training	NLST	*max.* 30,000	50.0	50.0
Validation	NLST	8889	20.6	79.4
Test	LIDC/IDRI	8582	13.3	86.7

in the area of medical imaging for lung. The database of CT images was obtained obtained through the contributions of seven academic centers and eight medical imaging companies.(Armato et al., 2004) Due to the large number of and variety in contributors, the dataset is relatively varied and contains both low-dose and full-dose CTs, taken with or without contrast, and the data was acquired with a wide range of scanner models and acquisition parameters. The CT images in the LIDC/IDRI database are accompanied by nodule annotations as well as subjective assessments of various nodule characteristics (such as suspected malignancy) provided by four expert thoracic radiologists. Unlike the NLST, the LIDC/IDRI database does not represent a screening population (*e.g.* exclusively long-term smokers), as the inclusion criteria allowed any type of participant.

All scans from the NLST and LIDC/IDRI datasets with an original slice thickness of at most 2.5 mm were processed by the same candidate generation model to provide center coordinates of potential nodules. These center coordinates were used to extract $12 \times 72 \times 72$ patches from the original scans, where each voxel represents $1.25 \times 0.5 \times 0.5$ mm of lung tissue. The candidate generation model itself was previously used in a high-scoring submission to the LUNA16 challenge by http://aidence.com, and can thus be assumed to be of sufficiently high accuracy and representative candidate generation of a real-world CAD application.

Values of interest for nodule detection lie approximately between -1000 Hounsfield Units (air) and 300 Hounsfield Units (soft-tissue) so this range was normalized to a $[-1, 1]$ range.

Due to the higher annotation quality and higher variety of acquisition types of the LIDC/IDRI dataset, along with the higher volume of available NLST image data, the training and validation is done on potential candidates (extracted by the candidate generation model) from the NLST dataset and testing is done on the LIDC/IDRI nodule candidates. This division of datasets, along with the exclusion of scans with a slice thickness greater than 2.5 mm, allowed us to use the reference standard for nodule detection as used by the LUNA16 grand challenge (Setio et al., 2016) and performance metric as specified by the ANODE09 study (Van Ginneken et al., 2010). This setup results in a total of 30,000 data samples for training, 8,889 for validation, and 8,582 for testing. Models are trained on subsets of this dataset of various sizes: 30, 300, 3000 and 30,000 samples. Each training set is balanced, and each smaller training set is a subset of all larger training sets. The details of the train, validation and test sets are specified in Table 1. It should be noted that for the NLST dataset, the training and validation set are blocked by patient; a patient (even with multiple CT scans from various timepoints and multiple nodules) will not appear in both datasets.

4.2. Network architecture & training procedure

A baseline network was established with 6 convolutional layers consisting of $3 \times 3 \times 3$ convolutions with 16, 32 and 64 filters, batch normalization and ReLU nonlinearities. In addition, the network uses 3D max pooling and a fully-connected layer.

The sequence of layers are visualised in Fig. 5. 3D Max pooling was done with same padding, and a filter size and stride of $1 \times 2 \times 2$ in the first instance to reduce only in the x- and y dimensions, and $2 \times 2 \times 2$ in the later two instances.

We minimize the cross-entropy loss using the Adam optimizer (Kingma and Ba, 2014). The weights were initialized using the uniform Xavier method (Glorot and Bengio, 2010). For training, we use a mini-batch size of 30 (the size of the smallest training set) for all training set sizes. We use validation-based early stopping (with patience 5). A single data augmentation scheme (continuous rotation by $0 - 360^o$, reflection over all axes, small translations over all axes, scaling between $0.8 - 1.2$, added noise, value remapping) was used for all training runs and all architectures (including the group convolutional architectures). The augmentations were applied at random to the training samples during training

We refer to the baseline network as the $\mathbb{Z}^3$-CNN, because, like every conventional 3D CNN, it is a G-CNN for the group of 3D translations, $\mathbb{Z}^3$. The $\mathbb{Z}^3$-CNN baseline, when trained on the whole dataset, was found to achieve competitive performance based on the LUNA16 grand challenge leader board, and therefore deemed sufficiently representative of a modern pulmonary nodule CAD system. The G-CNN architectures were created by simply replacing the convolution layers of the baseline with a group convolution of a group H.

4.3. Evaluation

Despite the availability of a clear definition of a lung nodule, given by the Fleischer Glossary (Hansell et al., 2008), several studies confirm that observers often disagree on what constitutes a lung nodule (Rubin et al., 2005; Armato III et al., 2007; 2009). This poses a problem in the benchmarking of CAD systems.

In order to deal with inter-observer disagreements, only those nodules accepted by 3 out of four radiologists (and $\geq$3mm and $\leq$30mm in largest axial diameter) are considered essential for the system to detect. Nodules accepted by fewer than three radiologists, those smaller than 3 mm or larger than 30 mm in diameter, or with benign characteristics such as calcification, are ignored in evaluation and do not count towards the positives or the negatives. The idea to differentiate between relevant (essential to detect) and irrelevant (optional to detect) findings was first proposed in the ANODE09 study (Van Ginneken et al., 2010).

ANODE09 also introduced the Free-Response Operating Characteristic (FROC) analysis, where the sensitivity is plotted against the average number of false positives per scan. FROC analysis, as opposed to any single scalar performance metric, makes it possible to deal with differences in preference regarding the trade-off between sensitivity and false positive rate for various users. We use this to evaluate our systems. To also facilitate direct quantitative comparisons between systems, we compute an overall system score based on the FROC analysis, which is the average of the sensitivity at seven predefined false positive rates ($\frac{1}{8}$; $\frac{1}{4}$; $\frac{1}{2}$; 1; 2; 4; and 8).

This evaluation protocol described in this section is identical to the method used to score the participants of the LUNA16 nodule detection grand challenge (Setio et al., 2016), and is the de facto standard for evaluation of lung nodule detection systems.[6]

An issue with this form of evaluation is that all nodules (big or small, subtle or easily spotted, malignant or benign) are weighted similarly. The primary motivation for nodule detection, however, is to find potentially malignant nodules. We therefore define a nodule to be *malignant* if at least three out of four radiologists suspected the nodule to be moderately or highly suspicious and no radiologists indicated the nodule to be moderately or highly unlikely to be malignant. In total, 129 nodules qualify as malignant according to this constraint.

[6] It should be noted that not all true nodules have been localised by the used candidate generation model and are therefore not in our set of potential candidates. The highest sensitivity that can be achieved is therefore 96.4%.

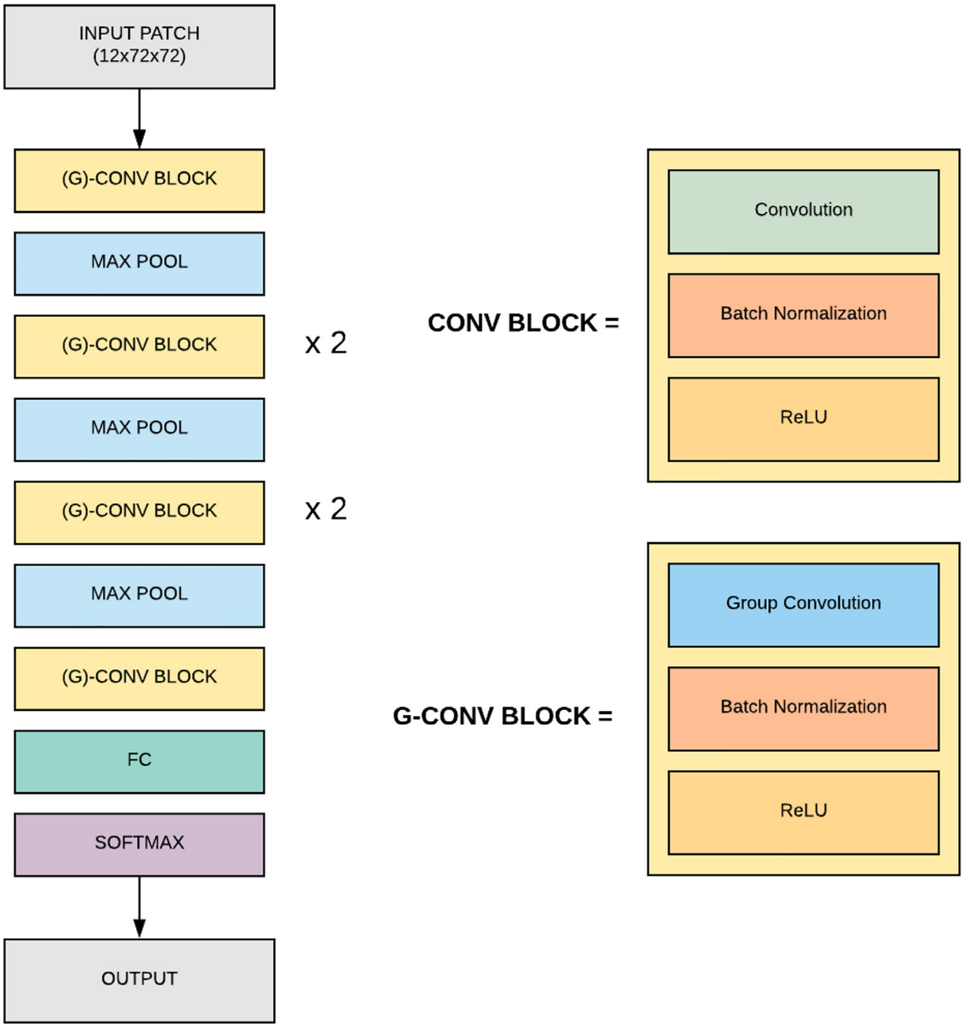


Fig. 5. Network architecture for baseline CNN and G-CNN.

5. Results

In this section we will present the results for the baseline and group convolutional networks for various training set sizes in terms of a FROC analysis and overall aggregated system score, as well as speed of convergence and an analysis of the sensitivity to malignant nodules. Scores were calculated by averaging test results over 3 training runs.

5.1. Nodule classification performance

This section presents the FROC curves and overall system scores by the baseline and the *G*-CNNs under various conditions. The highest (**bold**) and lowest (*italic*) scoring model per training set size will be highlighted.

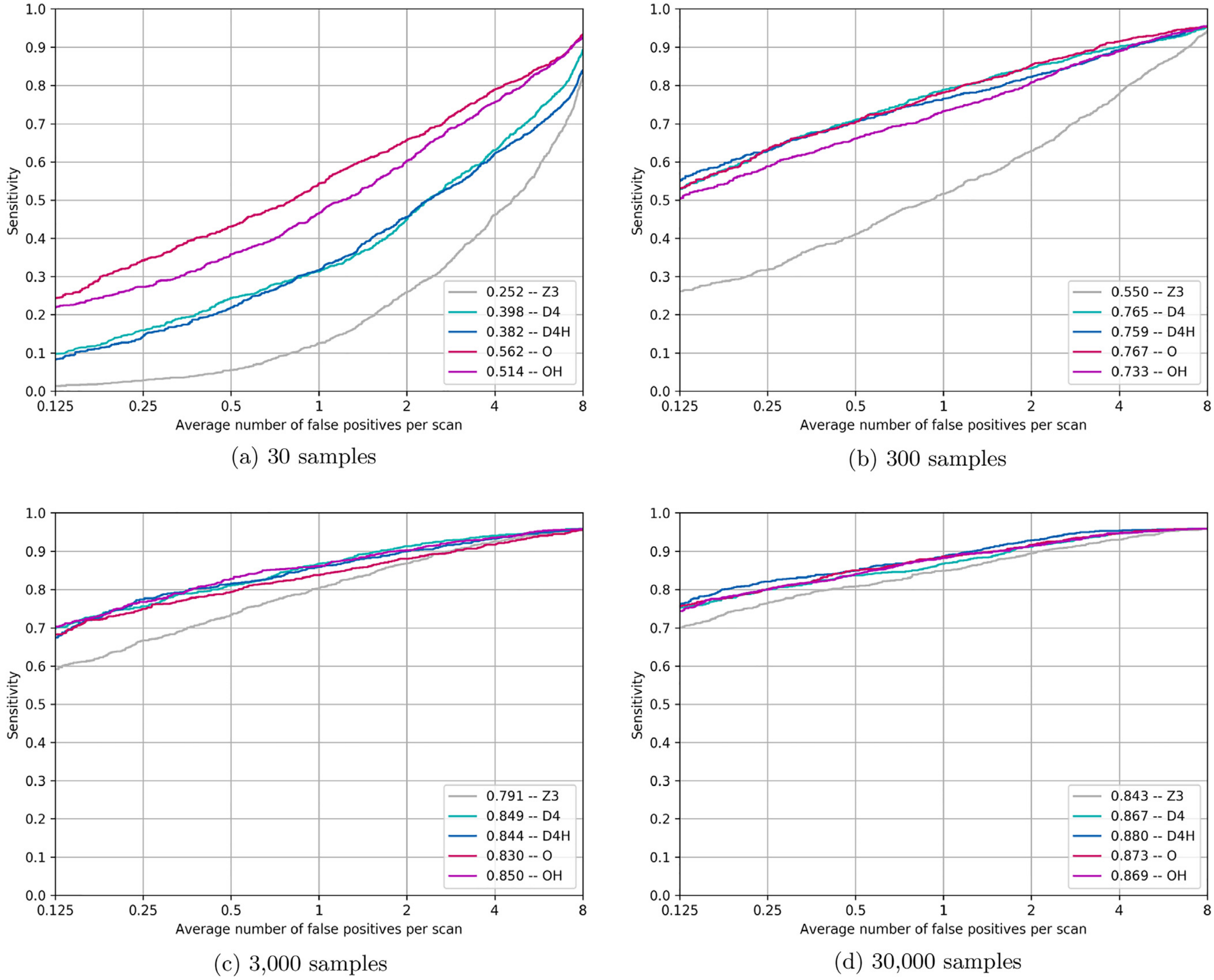
First, as described in Section 4, using a group convolution rather than a regular convolution with the same number of filters leads to an increase in number of 3D channels, and therefore an increase in number of parameters per filter. We want to keep the number of parameters for each model roughly similar in order to keep the networks comparable. Hence, the number of desired *G*-feature maps is calculated by dividing the number of filters for the baseline by $\sqrt{|H|}$, leading to $n_i/\sqrt{|H|}$ *G*-feature maps each with $|H|$ associated orientation channels. This was done for each group $H \in \{D_4, D_{4h}, O, O_h\}$ for every training dataset size and evaluated according to Section 4.3. These experiments also form the basis for the analyses with regards to convergence (see Section 5.3) and malignancy (see Section 5.4).

Table 2
Overall score for all training set sizes N and transformation groups G. The group $G = \mathbb{Z}^3$ corresponds to the standard translational CNN baseline.

N	$\mathbb{Z}^3$	D_4	D_{4h}	O	O_h
30	*0.252*	0.398	0.382	**0.562**	0.514
300	*0.550*	0.765	0.759	**0.767**	0.733
3,000	*0.791*	0.849	0.844	0.830	**0.850**
30,000	*0.843*	0.867	**0.880**	0.873	0.869

The overall system scores are presented in Table 2. For clarity, these results are visualised in Fig. 7, where the highest scoring *G*-CNN is positioned against the baseline CNN, which visually illustrates that the scores for the *G*-CNNs trained on training data set size *N* are more similar to the baseline trained on $10 \cdot N$ than *N*. Fig. 6 shows the FROC curve for each *G*-CNN and training set size.

As explained in Section 3.4, keeping the parameter budget roughly similar for the various models will still lead to an increase in computational cost for the group convolutional networks compared to the baseline, but we can enforce an equal compute. Table 3 lists the results of the baseline model, the results of the G-CNN with an equal parameter budget for two groups (*params*; also listed in Fig. 6) and an equal computational cost for these groups (*compute*). Although the networks with an equal compute budget



(a) 30 samples

(b) 300 samples

(c) 3,000 samples

(d) 30,000 samples

Fig. 6. FROC curves for all groups per training set size.

Table 3
Overall system score for all training set sizes N and transformation groups $\mathbb{Z}^3$, D_4 and D_{4h}. For the latter two, the number of output channels of the baseline are either divided by $\sqrt{|H|}$ to keep the number of parameters equal (center columns) or by $|H|$ to keep the compute equal (right columns).

	Baseline		Params		Compute
N	$\mathbb{Z}^3$	D_4	D_{4h}	D_4	D_{4h}
30	*0.252*	0.398	0.382	**0.424**	0.287
300	*0.550*	**0.765**	0.759	0.717	0.718
3,000	*0.791*	**0.849**	0.844	0.814	0.804
30,000	*0.843*	0.867	**0.880**	0.867	0.850

Table 4
Overall score for the baseline ($\mathbb{Z}^3$) and baseline with predictions averaged over the symmetry transformations ($\mathbb{Z}^3(D_4)$ and $\mathbb{Z}^3(D_{4h})$). For comparison, we also include the scores obtained by D_4 and D_{4h} G-CNNs (center columns).

			G-CNN		Test-time aug.
N	$\mathbb{Z}^3$	D_4	D_{4h}	$\mathbb{Z}^3$ (D_4)	$\mathbb{Z}^3$ (D_{4h})
30	*0.252*	**0.398**	0.382	0.314	0.286
300	*0.550*	**0.765**	0.759	0.551	0.590
3,000	*0.791*	**0.849**	0.844	0.821	0.830
30,000	*0.843*	0.867	**0.880**	0.858	0.848

Table 5
Number of epochs after which the loss is equal to or lower than the lowest validation loss achieved on the baseline for each group.

N	$\mathbb{Z}^3$	D_4	D_{4h}	O	O_h	total epochs
3000	*82*	33	22	21	**11**	*100*
30,000	*41*	4	9	7	**3**	*50*

perform worse than those with a comparable parameter budget, the G-CNNs still all outperform the baseline.

Note that this was only done for groups D_4 and D_{4h} for two reasons: first, the number of filters of the baseline are multiples of the order of these groups which allows for a neat division, and second, the orders of the cubic groups (24 and 48 for O and O_h resp.) are larger than the lowest number of filters (16), meaning a similar or lower compute cannot be achieved for a network of this size because we must have at least one filter per layer.

5.2. Test-time augmentation

It is common practice to apply various transformations to the testset at test time and average the predictions over these augmented versions of the test volume to achieve a single prediction

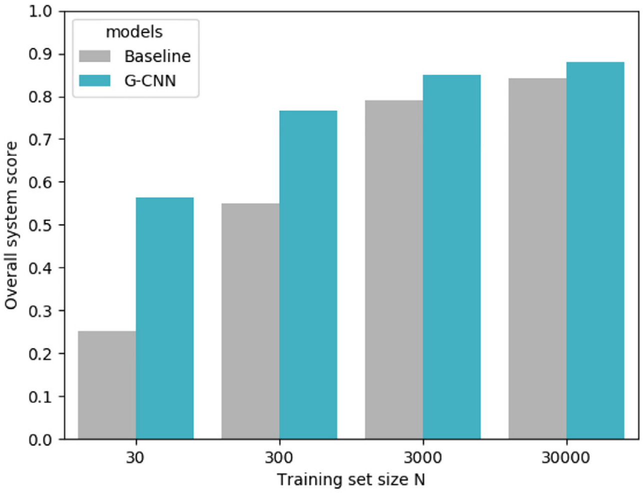


Fig. 7. Highest scoring *G*-CNN for each dataset size for all training set sizes *N* plotted against the baseline score.

for the patch. This typically boost performance and is often used as an approach to be somewhat invariant towards these transformations. Therefore, we want to compare this to our group convolutional approach.

The input data patches during test time are of a rectangular cuboid shape (12 × 72 × 72). This means all rectangular cuboid transformations can be applied to these input patches in a lossless manner, but not all cubic transformations. We therefore only compare with the set of rectangular cuboid transformations (the transformations of D_4 and D_{4h}) as augmentations at test time.

We apply these transformations to the volumes at test time for the baseline model. In Table 4, these are presented next to the results of the group convolutional models for these groups (based on an equal parameter budget). Although averaging over symmetries for the baseline model does result in an increase in performance over the baseline, the advantage is not nearly as big as using group convolutions for these symmetry groups instead.

5.3. Rate of convergence

Because of the increase in weight sharing, one might expect that a filter in *G*-CNN receives a stronger, less noisy gradient signal, leading to faster convergence. Fig. 8 plots the training loss per epoch, for training runs with dataset size 3000 and 30,000 and indeed show a sharper decline within the first few epochs for the group convolutional models. Table 5 lists the number of epochs it takes for a given network to reach a validation loss that is at least as good as the best validation loss of the baseline, which tends to be substantially lower for the *G*-CNNs.

5.4. Sensitivity to malignant nodules

The primary motivation for nodule detection is to find potentially malignant nodules, so we take a special interest in those nodules that can be considered malignant. We consider a set number of true positives from the testset that have the highest associated probability estimation relatively to the other true positives, and evaluate what number of these true positives is not only a nodule, but also malignant. These results are outlined in Table 6, where the number of malignant nodules in the set of n true positives that received the highest estimated probability is provided for each model, where $n \in \{100, 150, 250\}$. *E.g.* out of the 100 true positives with the highest probability estimation by the baseline model trained on 30,000 data samples, 13 were malignant.

Table 6
Number of malignant nodules from the n true positives that received the relatively highest probability estimation.

N	n	$\mathbb{Z}^3$	D_4	D_{4h}	O	O_h
	100	*5*	21	25	20	**37**
3,000	150	*6*	37	40	25	**45**
	250	*14*	59	59	51	**62**
	100	*13*	31	**45**	44	32
30,000	150	*18*	49	60	**61**	38
	250	*31*	61	**77**	**77**	63

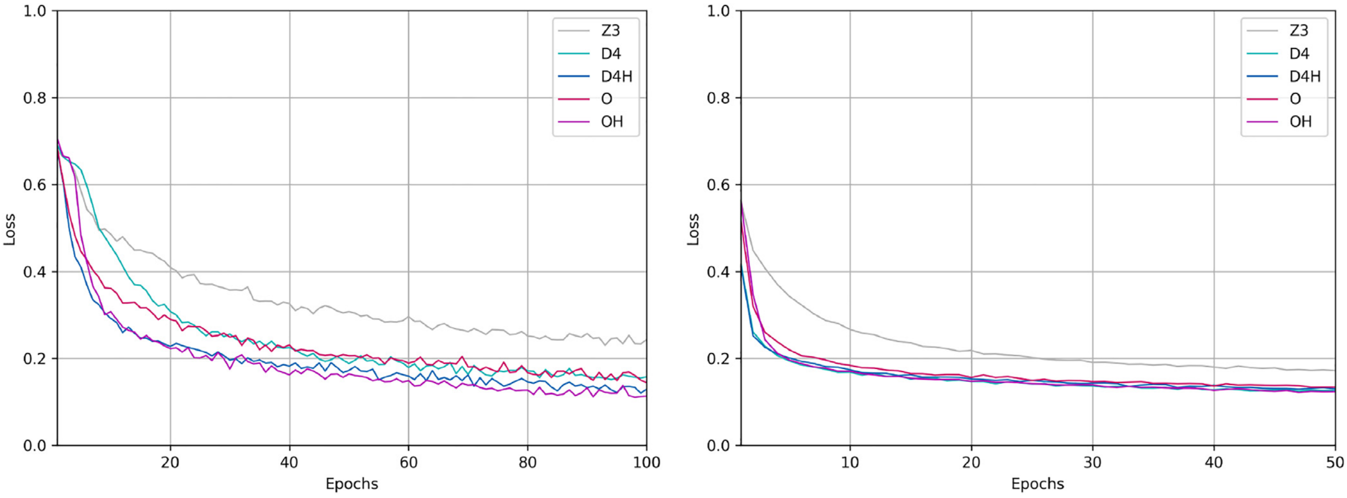
6. Discussion & future work

The general conclusion that can be drawn from our experiments is that G-CNNs consistently outperform CNNs on the task of lung nodule candidate classification. All tested G-CNNs outperform the baseline when trained on the same size dataset, both when the number of parameters is kept the same (and computational requirements are increased for G-CNNs) or when the computational budget is fixed (and the number of parameters of the G-CNN is reduced). This is true despite extensive data augmentation used in all experiments, and even when test-time data augmentation is used. Moreover, Table 2 and Fig. 7 illustrate that on a similar parameter budget, G-CNNs regularly outperform the baseline trained on 10 × more data. To take but one example, the *O*-CNN trained on N samples performs similarly to the baseline $\mathbb{Z}^3$-CNN trained on $10 \cdot N$ samples. Hence we can say that *G*-CNNs were found to be approximately 10 × more data efficient.

As the size of the training dataset is increased, the absolute improvement in performance brought about by the G-CNNs relative to the baseline is decreased. However, because increasing the dataset size also produces diminishing returns, the relative improvement brought about by G-CNNs is fairly constant, being approximately equivalent to an increase in dataset size by 10 × . The fact that the absolute improvements decrease with dataset size is not surprising, because in the limit of infinite data, the CNN should converge on the Bayes-optimal error rate (provided that it has sufficient capacity, and we are able to properly optimize it). What our experiments demonstrate is that G-CNNs converge to this limit at a significantly faster rate. Moreover, from a practical standpoint, what matters is not that we can squeeze out marginal performance increases at the very largest dataset sizes, but that (if our results hold in a more general context) the cost of data collection and annotation can be substantially reduced for future CAD efforts.

Similar to the CNN/G-CNN comparison, the difference between different G-CNNs (for different symmetry groups) appears to decrease as the dataset size is decreased. Hence, in the large data regime, a smaller group may be sufficient to boost performance, at lower computational cost. For smaller dataset sizes, we observed a notable advantage for the larger symmetry groups O and O_h (see Fig. 6a). This can be attributed to the trade-off between the number of distinct filters and the size of the group for a given parameter budget, as bigger groups allow for fewer distinct features given the same number of parameters. For a small dataset, there is insufficient data available to learn a large number of features in the first place, which may be why bigger groups appear to be more effective in this regime.

An additional benefit of G-CNNs, besides increased performance, is that they require fewer epochs to converge and generalise than regular CNNs. Fig. 8 shows that group convolutional models show a faster decline in train loss within the early stages of training, which can be attributed to fact that each parameter receives a gradient signal from multiple 3D feature maps at once. Additionally, Table 5 shows G-CNNs typically take only a fraction of the epochs required by the baseline to reach a validation loss that



(a) Train loss on 3,000 samples

(b) Train loss on 30,000 samples

Fig. 8. Learning curves for all networks trained on 3000 and 30,000 samples.

is better than the best validation loss achieved by the baseline. For example, the baseline $\mathbb{Z}^3$-CNN (trained on 30,000 samples) took 41 epochs to achieve its optimal validation loss, whereas the O_h-CNN reached the same performance in just 3 epochs. This could significantly reduce the time required for training. However, it should be noted that – as the G-CNN has more 3D channels given a fixed parameter budget – the processing of a single epoch takes longer for a G-CNN compared to a conventional CNN and sizes with the order of the group.

Lastly, an interesting observation for this particular application is that group convolution based models seem intrinsically more sensitive towards malignant nodules than regular CNNs. To illustrate this, there are more malignant nodule in the 100 most probable true positive nodules according to all G-CNNs than the number of malignant nodules in the 250 most probable nodules according to the baseline system (see Table 6). Another examples is that whereas there were 14 (out of 129) malignant nodules in the top 250 nodules for $\mathbb{Z}^3$ trained on 3000 data samples, there were more malignant nodules in the top 100 for any of the group convolutional models trained on the same dataset. Although the network was not trained on malignancy specifically (as a label), malignant nodules do tend to share certain characteristics that are indicators of malignancy such as a part-solid composition and/or marked spiculation that are under represented in the training set, as the majority of the nodules (in both the training- and testset) are benign. As we have already established group convolutional networks have a lower sample complexity, it is not surprising that they are also more capable of accurately classifying based on features that appear less often in the training set.

For the application of G-CNNs to the problem of pulmonary nodule classification, lower order groups of symmetry transformations of the rectangular cuboid, which was similar to the shape of our 3D input patch, were sufficient to significantly boost performance, speed of convergence and even proved more sensitive to malignant nodules, though at the cost of increased computational requirements. Fortunately, this issue can be resolved by further optimisations to the code, multi-GPU training, and future hardware advances.

Although pulmonary nodule classification may seem particularly suited for group convolutions given its symmetric nature, symmetries need not occur on a global level (as supported by results by Cohen and Welling (2016)) and can occur on small scales instead, which means we expect these results to generalise well to other applications.

7. Conclusion

In this work we have presented 3D Group Equivariant Convolutional Neural Networks (G-CNNs), and applied them to the problem of false positive reduction for lung nodule detection. 3D G-CNN architectures – obtained by simply replacing convolutions by group convolutions – unambiguously outperformed the baseline CNN on this task, especially on small datasets, without any further tuning. In our experiments, G-CNNs proved to be about 10 × more data efficient than conventional CNNs. This improvement in statistical efficiency corresponds to a major reduction in cost of data collection, and brings pulmonary nodule detection and other CAD systems closer to reality.

Conflict of interest

We declare that we have no conflicts of interest, financial or otherwise, that could bias the results of our research. In terms of reviewer conflicts, we would exclude reviewers from the University of Amsterdam, Qualcomm, Aidence, OpenAI and DeepMind, because we have recently worked at or collaborated with researchers at these institutions.

References

Al Mohammad, B., Brennan, P., Mello-Thoms, C., 2017. A review of lung cancer screening and the role of computer-aided detection. Clin. Radiol. 72 (1), 433–442.

American Cancer Society, 2017. Lung cancer detection and early prevention.Last revised: February 22, 2016.

American Cancer Society Statistics Center, 2017. Lung cancer key statistics. Last update: January 2017.

Armato, S.G., McLennan, G., McNitt-Gray, M.F., Meyer, C.R., Yankelevitz, D., Aberle, D.R., Henschke, C.I., Hoffman, E.A., Kazerooni, E.A., MacMahon, H., Reeves, A.P., Croft, B.Y., Clarke, L.P., 2004. Lung image database consortium research group. lung image database consortium: developing a resource for the medical imaging research community. Radiology 232 (3), 739–748.

Armato III, S.G., McLennan, G., Bidaut, L., McNitt-Gray, M.F., Meyer, C.R., Reeves, A.P., Zhao, B., Aberle, D.R., Henschke, C.I., Hoffman, E.A., et al., 2007. The lung image database consortium (lidc): an evaluation of radiologist variability in the identification of lung nodules on ct scans.. Acad. Radiol. 14 (11). 1409–21. URL: http://www.ncbi.nlm.nih.gov/pubmed/17964464 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2290739. doi: 10.1016/j.acra.2007.07.008.

Armato III, S.G., Roberts, R.Y., Kocherginsky, M., Aberle, D.R., Kazerooni, E.A., MacMahon, H., van Beek, E.J., Yankelevitz, D., McLennan, G., McNitt-Gray, M.F., et al., 2009. Assessment of radiologist performance in the detection of lung nodules: dependence on the definition of "truth". Acad. Radiol. 16 (1), 28–38.

Bekkers, E. J., Lafarge, M. W., Veta, M., Eppenhof, K. A. J., Pluim, J. P. W., 2018. Roto-Translation covariant convolutional networks for medical image analysis. Unknown.

Bhargavan, M., Kaye, A., Forman, H., Sunshine, J., 2009. Workload of radiologists in united states in 2006–2007 and trends since 1991–1992. Radiology 252 (2), 458–467.

Bogoni, L., Ko, J.P., Alpert, J., Anand, V., Fantauzzi, J., Florin, C.H., Koo, C.W., Mason, D., Rom, W., Shiau, M., et al., 2012. Impact of a computer-aided detection (cad) system integrated into a picture archiving and communication system (pacs) on reader sensitivity and efficiency for the detection of lung nodules in thoracic ct exams.. J. Digit. Imaging 25 (6).

Cohen, T., Welling, M., 2016. Group equivariant convolutional networks. In: Proceedings of the International Conference on Machine Learning, pp. 2990–2999.

Cohen, T.S., Geiger, M., Koehler, J., Welling, M., 2018. Spherical CNNs. In: Proceedings of the International Conference on Learning Representations (ICLR).

Cohen, T.S., Welling, M., 2017. Steerable CNNS. In: Proceedings of the International Conference on Learning Representations.

Dieleman, S., Fauw, J., Kavukcuaglu, K., 2016. Exploiting cyclic symmetry in convolutional neural networks. Proceedings of the International Conference on Machine Learning, 1889–1898.

Firmino, M., Morais, A., Medoca, R., Dantas, M., Hekis, H., Valentim, R., 2014. Computer-aided detection system for lung cancer in computed tomography scans: review and future prospects. Biomed. Eng. Online 13 (1).

van Ginneken, B., 2017. Fifty years of computer analysis in chest imaging: rule-based, machine learning, deep learning. Radiol. Phys. Technol. 10 (2).

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the International Conference on Artificial Intelligence and Statistics, 249–256.

Hansell, D., Bankier, A., MacMahon, H., McLoud, T., Muller, N., Remy, J., 2008. Fleischner society: glossary of terms for thoracic imaging. Radiology 697–722.

Kingma, D., Ba, J., 2014. Adam: a method for stochastic optimization. CoRR.

Kondor, R., Son, H. T., Pan, H., Anderson, B., Trivedi, S., 2018. Covariant compositional networks for learning graphs. unknown.

Lauritzen, P.M., Andersen, J.G., Stokke, M.V., Tennstrand, A.L., Aamodt, R., Heggelund, T., Dahl, F.A., Sandbæk, G., Hurlen, P., Gulbrandsen, P., 2016. Radiologist-initiated double reading of abdominal ct: retrospective analysis of the clinical importance of changes to radiology reports.. BMJ Qual. Safety 25 (8), 595–603. URL: http://www.ncbi.nlm.nih.gov/pubmed/27013638 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4975845. doi: 10.1136/bmjqs-2015-004536.

McNitt-Gray, M.F., Armato, S.G., Meyer, C.R., Reeves, A.P., McLennan, G., Pais, R.C., Freymann, J., Brown, M.S., Engelmann, R.M., Bland, P.H., Laderach, G.E., Piker, C., Guo, J., Towfic, Z., Qing, D.P.Y., Yankelevitz, D.F., Aberle, D.R., van Beek, E.J.R., MacMahon, H., Kazerooni, E.A., Croft, B.Y., Clarke, L.P., 2007. The lung image database consortium (LIDC) data collection process for nodule detection and annotation. Radiology 14 (12), 1464–1474.

National Lung Screening Trial, 2011. Reduced lung-cancer mortality with low-dose computed tomographic screening. N Top N. Engl. J. Med. 365 (5), 395–409.

Oudkerk, M., Devaraj, A., Vliegenthart, R., Henzler, T., Prosch, H., Heussel, C.P., Bastarrika, G., Sverzellati, N., Mascalchi, M., Delorme, S., et al., 2017. European position statement on lung cancer screening. Lancet Oncol. 18, 754–766. doi:10.1016/S1470-2045(17)30861-6.

Rubin, G.D., Lyo, J.K., Paik, D.S., Sherbondy, A.J., Chow, L.C., Leung, A.N., Mindelzun, R., Schraedley-Desmond, P.K., Zinck, S.E., Naidich, D.P., et al., 2005. Pulmonary nodules on multi–detector row ct scans: performance comparison of radiologists and computer-aided detection. Radiology 234 (1), 274–283.

Setio, A.A.A., Traverso, A., De Bel, T., Berens, M.S., van den Bogaard, C., Cerello, P., Chen, H., Dou, Q., Fantacci, M.E., Geurts, B., et al., 2016. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. CoRR. 1612.08012. abs/1612.08012. URL: http://arxiv.org/abs/1612.08012.

Simard, P., Steinkraus, D., Platt, J., 2003. Best practices for convolutional neural networks applied to visual document analysis. In: Proceedings of the ICDAR, 3, pp. 958–962.

Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., Riley, P., 2018. Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds.

Van Ginneken, B., Armato, S.G., de Hoop, B., van Amelsvoort-van de Vorst, S., Duindam, T., Niemeijer, M., Murphy, K., Schilham, A., Retico, A., Fantacci, M.E., et al., 2010. Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: the anode09 study. Med. Image Anal. 14 (6), 707–722.

Wang, H., Naghavi, M., Allen, C., Barber, R.M., Bhutta, Z.A., Carter, A., Casey, D.C., Charlson, F.J., Chen, A.Z., Coates, M.M., et al., 2016. Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the global burden of disease study. Lancet 388, 1459–1544. PMID 27733281. doi: 10.1016/S0140-6736(16)31012-1.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., Cohen, T., 2018a. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. arXiv: 1807.02547.

Weiler, M., Hamprecht, F.A., Storath, M., 2018b. Learning steerable filters for rotation equivariant CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Wormanns, D., Ludwig, K., Beyer, F., Heindel, W., Diederich, S., 2005. Detection of pulmonary nodules at multirow-detector ct: effectiveness of double reading to improve sensitivity at standard-dose and low-dose chest ct. Eur. J. Radiol. 15 (1), 14–22.

Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J., 2017. Harmonic networks: deep translation and rotation equivariance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Zhao, Y., de Bock, G.H., Vliegenthart, R., van Klaveren, R.J., Wang, Y., Bogoni, L., de Jong, P.A., Mali, W.P., van Ooijen, P.M., Oudkerk, M., 2012. Performance of computer-aided detection of pulmonary nodules in low-dose ct: comparison with double reading by nodule volume.. Eur. Radiol. 22 (10). 2076–84. URL: http://www.ncbi.nlm.nih.gov/pubmed/22814824 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3431468. doi: 10.1007/s00330-012-2437-y.