Spatial Transforming Network for chest X-Ray images preprocessing

Andrey Galichin [1] **Evgeny Gurov** [1] **Arkadiy Vladimirov** [1]

Abstract

In this project we try to solve the problem of unsupervised chest X-Ray images alignment. We believe that proper alignment of medical images may improve accuracy of diseases classification. To solve this problem we use Style Transfer approach in combination with Spatial Transformer Network achitecture, which shows quite satisfactory results.

Github repo: https://github.com/bizzare-hub/Chest-Xray-alignment-using-STN.git
Presentation: https://github.com/bizzare-hub/Chest-Xray-alignment-using-STN/blob/main/presentation.pdf

1. Introduction

Chest X-ray imaging is a widely used diagnostic tool for the detection and monitoring of various medical conditions, including pneumonia, tuberculosis, lung cancer etc. The accurate analysis and comparison of chest X-ray images can be challenging due to inconsistencies in the image orientation, scale, and position. These misalignments can adversely affect the performance of downstream tasks such as disease classification, segmentation, and other computer-aided diagnosis systems. In recent years, there has been a growing interest in developing machine learning-based techniques to preprocess and align chest X-ray images for improved diagnostic performance.

In this work, we implement a novel preprocessing framework that leverages a neural network architecture combining a ResNet18 encoder and a Spatial Transformer block for automatic alignment of chest X-ray images. Due to the absence of labeled data we use consistency loss, first introduced in Style Transfer approach, for training this architecture in unsupervised manner.

The main contributions of this report are as follows:

We implement a novel preprocessing framework for automatic alignment of chest X-ray images, which consists of a neural network architecture that combines a ResNet18 encoder with a Spatial Transformer block and consistency loss. We properly tune the parameters of the algorithm to achieve good performance and prevent divergence. We provide evaluation of the improvement of quality for such downstream task as classification of fourteen diseases in ChestX-ray14 dataset.

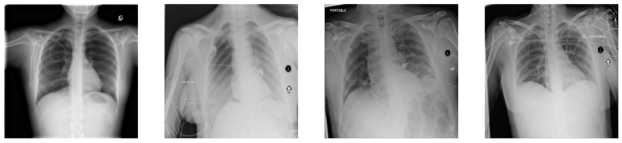

Figure 1. Examples of not aligned images in ChestX-ray14

2. Related work

At the moment, the approach we have taken as a basis from the article (Liu et al., 2019) in fact has no analogues. As an alternative scheme, in theory it is possible to apply all kinds of affine transforms as augmentations of the dataset, in order to make the model which solves the downstream problem invariant to such transformations. For example, ChexNet (Rajpurkar et al., 2017) introduces random horizontal flipping as an augmentation to make model invariant to such transformation. Chexclusion (Seyyed-Kalantari et al., 2020) improves the initial scheme, adding random rotation to the augmentations set. Rotation range is tuned using a standard cross validation scheme. Nevertheless, we tend to believe that the approach we have chosen is more universal and can show better results. Furthermore, the method we have chosen is able to crop the edges of images that do not contain X-rays, which increases the information content of each individual image. The basis of our method is the Spatial Transformer architecture described in (Jaderberg et al., 2016). Training with this approach is based on the concept of Style Transfer (Johnson et al., 2016) and the final pipeline is essentially a combination of these two ideas.

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Andrey Galichin <Andrey.Galichin@skoltech.ru>.

Final Projects of the Machine Learning 2023 Course, Skoltech, Moscow, Russian Federation, 2023.

3. Algorithms and Models

Full and reproducible code for the pipeline described below may be found at the project's repo.

3.1. The architecture

As we have already said, our goal is to achieve the perfect alignment of the input Chest X-ray image by regressing the corresponding affine transform parameters. To correctly obtain them, we build the network on the idea of spatial transformer block, first introduced in (Jaderberg et al., 2016). The idea is to use a simple localization network consisting of several linear layers to regress the affine matrix parameters, and then apply a warping transform to the initial image to obtain its aligned variant. Because in our formulation we do not have a training dataset for supervised learning, we follow the idea stated in (Liu et al., 2019). Specifically, we align all the images to a single target image, which we call "Canonical chest". To obtain it, we randomly sample 1000 images from ChestX-ray14 dataset and average them. After that, we also crop out the central view tightly bounding the two lungs.

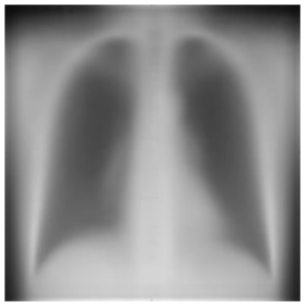

Figure 2. Canonical chest representation

3.1.1. Alignment module

After obtaining the canonical chest as the target image, we frame the transformation learning as minimizing the structure divergence between the transformed image and the target image. Let I and T denote the transformed input image and the target image respectively. Given I, and the alignment network ϕ_A, we obtain the transformed image $\phi_A(I)$. To let $\phi_A(I)$ have desired structure, we minimize the structure loss

$$L_s = f(\phi_A(I), T). \tag{1}$$

Specifically, we use a light-weighted ResNet18 as the backbone of ϕ_A plus two fully-connected layers. The output of the alignment network is a group of five parameters $(t_x, t_y, s_x, s_y, \theta)$ of the affine transformation. Here t_x and t_y stand for horizontal and vertical displacements, s_x and s_y stand for horizontal and vertical scaling, θ stands for the rotation angle. After all, the transformation $\phi_A(I)$ represents in the following way:

$$\phi_A(I) = B\left(\begin{pmatrix} s_x\cos\theta & -s_y\sin\theta t_x \\ s_x\sin\theta & s_y\cos\theta t_y \end{pmatrix} G(I), I\right), \tag{2}$$

where B stands for a bilinear interpolating function, and G for a regular grid function.

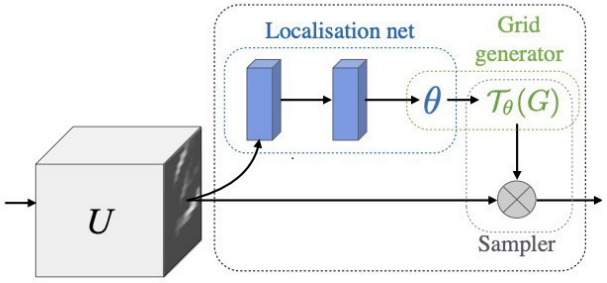

Figure 3. Spatial transformer architecture

3.1.2. Losses

To encourage $\phi_A(I)$ to have similar structure with T without annotated data, as an alternative, we propose to use the perceptual loss (Johnson et al., 2016), and, specifically, its feature reconstruction loss part, in order to preserve the content structure.

$$L_{\text{feat}}(\phi_A(I), T) = \frac{||N_{\text{feat}}(\phi_A(I)) - N_{\text{feat}}(T)||_2}{CHW}. \tag{3}$$

Here N_{feat} stands for some feature extraction convolutional network, which is VGG16 in our case. We also include the consistency loss $L_{\text{pixel}}(\phi_A(I), T)$ to preserve the initial image, which is actually just a metric generated by the second Euclidean norm. However, during experiments, we met some problems with stable convergence of alignment model. The problem was hidden in the fact, that pretrained feature extraction network is learned to process the images, while at the same time we passed to it our canonical chest, which is in fact some other entity. To this end, we proposed additionally to T also collect *Canonical features*, which we denote T_{feat}. This is an average of randomly drawn n images' feature maps obtained by passing them through our feature extraction network.

$$T_{\text{feat}} = \sum_{k=1}^{n} N_{\text{feat}}(I_k) \tag{4}$$

Then, the corresponing feature reconstruction loss representation changes — N_{feat} is replaced with T_{feat}.

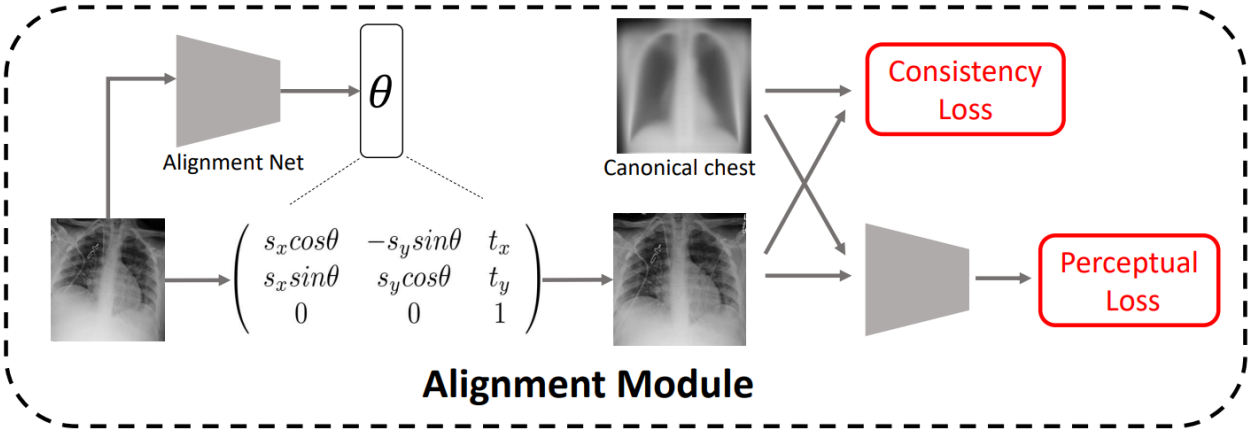

Figure 4. First, we take as input a chest X-ray image, which is passed threw our alignment model, which consists of ResNet18 and STN block. From it, we obtain affine transform parameters, which we use to warp the initial image and obtain its aligned variant. Then, we use pretrained feature extractor (VGG16) to obtain the features maps to calculate L_{feat} and L_{pixel}.

The final loss is a combination of two loss functions, where λ is a weight to balance the impact of the consistency loss:

$$L = L_{\text{feat}} + \lambda L_{\text{pixel}}. \tag{5}$$

3.1.3. Overall architecture

First, we take as input a chest X-ray image, which is passed threw our alignment model, which consists of ResNet18 and STN block. From it, we obtain affine transform parameters, which we use to warp the initial image and obtain its aligned variant. Then, we use pretrained feature extractor (VGG16) to obtain the features maps to calculate L_{feat} and L_{pixel}.

3.2. The dataset

For training and evaluation we use ChestX-ray14 dataset (Wang et al., 2017), which contains 112120 chest X-ray images of 30805 unique patients. Each radiography is labeled with one or multiple types of 14 diseases: Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Edema, Emphysema, Fibrosis, Pleural Thickening and Hernia. Original images are stored in grayscale format and have size 1024x1024.

3.3. Preprocessing

To train our alignment network, we don't need the actual labels, but only Chest X-rays, so we omit them during our dataset pipeline construction.

Our pipeline contains ResNet18 and VGG16 networks pretrained on the ImageNet dataset (Deng et al., 2009) and expect (3, 320, 320) data as input. Therefore, we shrink the original image of size 1024x1024 to be of size 320x320 and convert single channel X-ray images into 3-channel RGB images by simple tiling of the channel axis.

During preprocessing, we normalize the image by scaling it to [0; 1] range. As for augmentations, we randomly shift, scale rotate the image by maximum value of 10 degrees to enrich the disalignment variations in our dataset. Also standard color augmentations such as brightness and contrast are applied. Finally, we normalize image by mean and std from ImageNet dataset.

As visually satisfactory results on the whole dataset would already may be considered as a success, and moreover any quantitative evaluations of the results are questionable, we haven't performed any splitting of our dataset, limiting our validation to online evaluation of the transformed images during training. We visualize 1 random result of our model work once every 100 steps.

3.4. Training process

The weights of the networks are initialized with weights pretrained on ImageNet. The weight of reconstruction loss is set to 20. The end-to-end model is trained by AdamW optimizer with standard parameters. We set the batch size 8, initial learning rate 1e-4 and weight decay 1e-4. The learning rate starts from 1e-4 and decreases linearly to 1e-6 at the end of learning process. The training procedure is done in 5 epochs in total. One epoch takes approx. 7000 forward-backward steps. With our setup (2 Nvidia Geforce GTX 1080 TI, 11gb) it required 1.5h to train the model.

Configuration	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema
DenseNet121 (standard)	**0.7801**	0.8666	0.7434	0.8315	**0.8314**	**0.9286**
Densenet121 (ours)	0.7787	**0.8692**	**0.7501**	**0.8481**	0.8304	0.9231

Fibrosis	Hernia	Infiltration	Mass	Nodule	Pleural Thickening	Pneumonia	Pneumothorax	Mean
0.8282	0.8918	**0.6954**	0.8224	0.7578	**0.7867**	0.7102	**0.8816**	0.8110
0.8371	**0.9064**	0.6891	**0.8347**	**0.7712**	0.7833	**0.7309**	0.8619	**0.8153**

Figure 5. Classification ROC AUC

4. Experiments and Results

4.1. Visualizations

Here we present some examples of our Alignment network on ChestX-ray14 dataset ("Initial" is an input chest, "Aligned" is the aligned version outputted by our model):

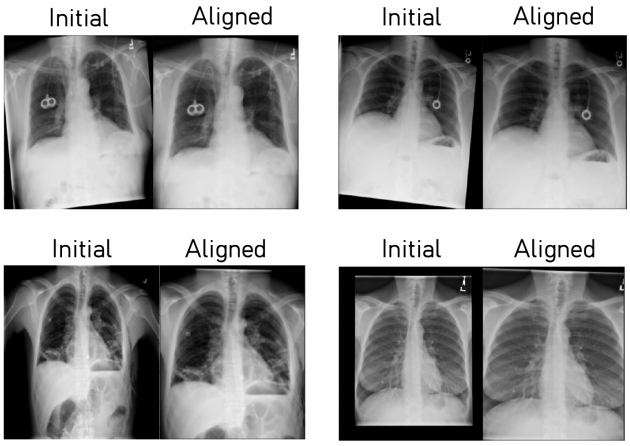

Figure 6. Alignment results visualization

4.2. Classification

In addition to the visualizations, we also evaluated our model performance on downstream classification task. Specifically, we constructed a standard training pipeline following CheXNet paper (Rajpurkar et al., 2017) and performed 2 experiments on different datasets: standard Chest X-ray14 and it's aligned version, created using our alignment model.

To train and test our models we use a public data split: 86524 and 25596 chest X-ray images for training and evaluation purposes. Data split is performed on patient level, which means all patient images are located only in one of the splits.

We formulate the Chest X-ray disease recognition as multi-label classification problem. Our network outputs a 14-dimensional vector indicating a positive probability for each kind of listed diseases. As an objective we use standard binary cross entropy loss, calculated for each class separately and averaged after t produce a single scalar.

The results on all 14 classes are shown in the tables, metric is ROC AUC (Receiver Operating Charachteristic Area Under Curve).

From the results we can see that our alignment module does improve the classification model results on 8 out of 14 diseases.

5. Conclusion

In this report, we have presented a novel preprocessing framework for automatic alignment of chest X-ray images, aimed at improving the performance of downstream tasks such as disease classification, segmentation, and others. Our approach leverages a neural network architecture that combines a ResNet18 encoder and a Spatial Transformer block, enabling efficient and robust alignment of input images.

The results of experiments we have conducted show visually pleasing and promising results, providing satisfactory grounds for future development of the method, which, compared to classical augmentation, offers several advantages including adaptability to complex transformations, data compression and absorbing computational costs of learning variously transformed data that allows potentional reduce of classification models complexity, making it a valuable preprocessing tool.

As future work, we plan to explore the integration of additional deep learning techniques to further improve the alignment process and investigate the applicability of our framework to other medical imaging modalities. Additionally, we aim to develop end-to-end models that combine image alignment and downstream tasks in a single archi-

tecture to further optimize performance and streamline the diagnostic process.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial transformer networks. *arXiv preprint arXiv:1506.02025v3*, 2016.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155v1*, 2016.

Liu, J., Zhao, G., Fei, Y., Zhang, M., Wang, Y., and Yu, Y. Align, attend and locate: Chest x-ray diagnosis via contrast induced attention network with limited supervision. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10631–10640, 2019. doi: 10.1109/ICCV.2019.01073.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Ball, R. L., Langlotz, C., Shpanskaya, K., Lungren, M. P., and Ng, A. Y. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225v3*, 2017.

Seyyed-Kalantari, L., Liu, G., McDermott, M., Chen, I. Y., and Ghassemi, M. Chexclusion: Fairness gaps in deep chest x-ray classifiers. *arXiv preprint arXiv:2003.00827v2*, 2020.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summer, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *arXiv preprint arXiv:1705.02315v5*, 2017.

A. Team member's contributions

Andrey Galichin (60% of work)

- Reviewing literature on the topic
- Coding the main pipeline
- GitHub Repo Support

Evegeny Gurov (20% of work)

- Performing experiments
- Presentation design
- Writing the report

Arkadiy Vladimirov (20% of work)

- Performing experiments
- Presentation design
- Writing the report

B. Reproducibility checklist

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☐ Yes.
   ☑ No.
   ☐ Not applicable.

   Students' comment: None

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☐ Yes.
   ☑ No.
   ☐ Not applicable.

   Students' comment: A reference to the article (Wang et al., 2017) about the dataset itself with a link to it is provided instead.

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: None

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: Due to the complexity of the model and the amount of data automatic search for the best hyper-parameter configuration is not quite applicable.

9. The exact number of evaluation runs is included.

   ☐ Yes.
   ☐ No.
   ☑ Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.

    ☐ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None

12. Clearly defined error bars are included in the report.

    ☐ Yes.
    ☐ No.
    ☑ Not applicable.

    Students' comment: None

13. A description of the computing infrastructure used is included in the report.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: None