Identify and extract text from photos of DICOM images for teleradiology

Bob Warfsman, Wanfei Luo

1. Introduction

In recent years, as the development of image processing technology improves and machine learning techniques emerge, medical imaging has become more common and informative. Medical image files generally are stored in a picture archival and communications system (PACS) by conforming images to the Digital Imaging and Communications in Medicine (DICOM) standard -- the international standard to process, store, retrieve, and display medical imaging information. Unfortunately, information exchange and sharing can be difficult, since different hospitals, sometimes even different departments use different PACS infrastructures to maintain and process medical imaging data. This process can be problematic especially for radiologists and patients who have the demand for image transformation across different PACS systems.

To improve data sharing capabilities, using smartphone photographs as proxies for DICOM images has become a new possibility. Radiologists use a smartphone to capture the displayed DICOM image on the medical monitor, and the photographic image can be transferred to an image that is close to the DICOM image by using autoencoder. This approach will bring in interoperability benefits for institutions or radiologists with different PACS infrastructures that help them to provide timely services and better patient care.

Displayed DICOM format image consists of a medical image and a series of personal identifiable information – PII, regarding patient demographics, diagnostic parameters, etc. DICOM users are responsible for protecting the medical data of their patients that all information that can be used to identify the patients must be removed before images are exacted from other devices for any purposes (DICOM.org). To protect patients' rights in the aspects of privacy and safety, an image preprocessing has to be considered and implemented when using photographs as proxies to conduct image sharing process. Therefore, it is highly desirable to design an application that can automatically anonymize PII from a photograph of a medical screen so that the medical image can be anonymized to share via smartphone with other radiologists who are out-of-network. This project aims to implement supervised learning to perform DICOM image text detection and extraction for teleradiology. The two main objectives of this project are identifying the ultrasound machine from a photograph of the screen – part 1 and detecting and capturing plain text from photographs – part 2.

2. Literature Review

To obtain a deep understanding of the project subject, mostly related to text extraction and glare detection, a total of six research papers have been reviewed. Three of the six peer-reviewed papers which are most related to the project will be thoroughly reviewed below.

Deep learning for text spotting

Max Jaderberg, a researcher from the University of Oxford, discussed the topic of text spotting in his paper in 2014. The paper mainly focuses on two approaches of text spotting, character-centric text spotting and word-centric text spotting.
Character-centric text spotting is an end-to-end text spotting system that essentially treats characters as the atomic building blocks for text (Jaderberg, 2014). In his approach, Maxout is used as the non-linear activation functions within the network, instead of using traditional sigmoid or RuLU functions. Maxout

allows the CNN to model multiple modes of data as taking the maximum responses over a mixture of linear functions (Jaderberg, 2014). The training process consists of two stages. In the first stage, a case-insensitive CNN character classifier is learned. In the second stage, the feature extractors are applied to other classification problems. Therefore, a total of four "state-of-the-art CNN classifiers" have been derived: a character/background classifier, a case-insensitive character classifier, a case-sensitive character classifier, and a bigram classifier (Jaderberg, 2014). Similar to other approaches, this approach also starts by computing a text saliency map. More specifically, a total of 16 scales are involved in this process to target text heights between 16 and 260 pixels by resizing the input image as CNN is trained to detect text at a single standardized height (Jaderberg, 2014). After obtaining the saliency map, the next step is to identify line of text where a probability map is generated and to define local regions with high probability. Then, he proposed to split text lines into words that the bounding boxes are filtered based on constraints like box height, aspect ratio, etc and sorting them based on per-pixel text saliency score.

Scene Text Detection and Segmentation Based on Cascaded Convolution Neural Networks

In this paper, the two scholars proposed to use a CNN-based method – segmentation network, SNet for scene text extraction and segmentation by using both the edges and the whole regions of text. After inputting the images, the first step of their method is text-aware CTR extraction; a deeply supervised CNN network, DNet, a fully convolutional network, is designed to predict the saliency for each pixel in order to detect the initial location of text (Tang and Wu, 2017). The shape of the text regions provide extremely important information in scene text extraction for distinguishing text and background, since in CNN the edges can be seen as local information and the region can be seen as global information that two of them can be used as the supervisory information of the shallow and also the deep layers of the CNN during model training (Tang and Wu, 2017). The second step of their method is CTR refinement to further refine the text region segmentation results which is necessary to images that contain background (Tang and Wu, 2017). After CTR refinement, the third step is CTR classification which is a two-class problem in image classification. The model is based on VGGNet-16 where the first three blocks remain and the rest of the blocks are cut to build CNet instead. When dealing with the size of input images, all images are fixed as 32 in height and corresponding width for each individual image (Tang and Wu, 2017). The performance of text detection achieves the highest recall (0.859) and F-measure (0.880) compared with other approaches using the same dataset (Tang and Wu, 2017). However, when the background has a very close color to the text or very similar size, the performance of the method gets worse; also, the strong light and non-uniform illumination also have negative impacts on the detection results(Tang and Wu, 2017).

Comparison of medical image classification accuracy among three machine learning methods

The paper evaluates the accuracy of medical image classification among machine learning methods which include Support Vector Machine (SVM), Artificial Neural Network (ANN), and Convolution Neural Network (CNN) by using DICOM format and JPEG format images.
For SVM and ANN classifier, commonly used features – median, entropy, area, contrast, energy, and homogeneity are used to show features of the image, which the area feature implies the total area obtained as the all images are binarized (Maruyama, T., et al.,2018). All the images are labeled as "CT", "MRI" or "X-ray". In their study, ANN is three-layered which consists of one input layer with 6 input units, one hidden layer with 10 hidden units, and one output layer with 3 output units plus back propagation learning algorithm (Maruyama, T., et al.,2018). A pre-trained AlexNet and support vector machine is used during transfer learning for CNN classifier. As a result, CNN shows an outstanding performance to identify medical images that not only maintains high precision of the images but also shortens the overall time period (Maruyama, T., et al.,2018). The study demonstrates that compared to conventional machine

learning methods, CNN is the most effective classifier for computer-aided medical image processing and inspection system no matter the image format is either DICOM or JPEG.

3. Data

3.1 Original Dataset

The data that is used for this project comes from a breast cancer screening study. The study involved yearly screenings of women with dense breast tissue and women with an increased risk of breast cancer. The study was conducted by performing yearly screening over the course of 5 years.

The dataset consists of ~27K DICOM files from a variety of ultrasound machines taken from several different hospitals. The DICOM files used in this project consist of many different metadata attributes and a single data payload from a single ultrasound machine. Among the attributes are patient ID, (anonymized for research), date and time information, location, ultrasound machine model, and several pieces of information about the data payload itself. The actual data in the file may be a video file, single channel image, or a three channel image. All of the images were stored using a lossless file format so there is no compression in the images. The average size of each DICOM file is ~370KB.

For this project, only the data payload was used, specifically data payloads that were 3-channel ultrasound images. The dataset collected in different ultrasound machine manufactures is utilized, consisting of 6 different types of machine shown in figure 1. The ultrasound machines used in this project were two from GE, one from Phillips, two from Siemens and one from HDI. There are many additional ultrasound machines used in the industry today. While this is only a small sample, we believe it will be very easy to add any additional machine to this project with a very limited number of samples. It is unknown how the accuracy will be affected if this model tried to predict 100 machines versus the six we have.

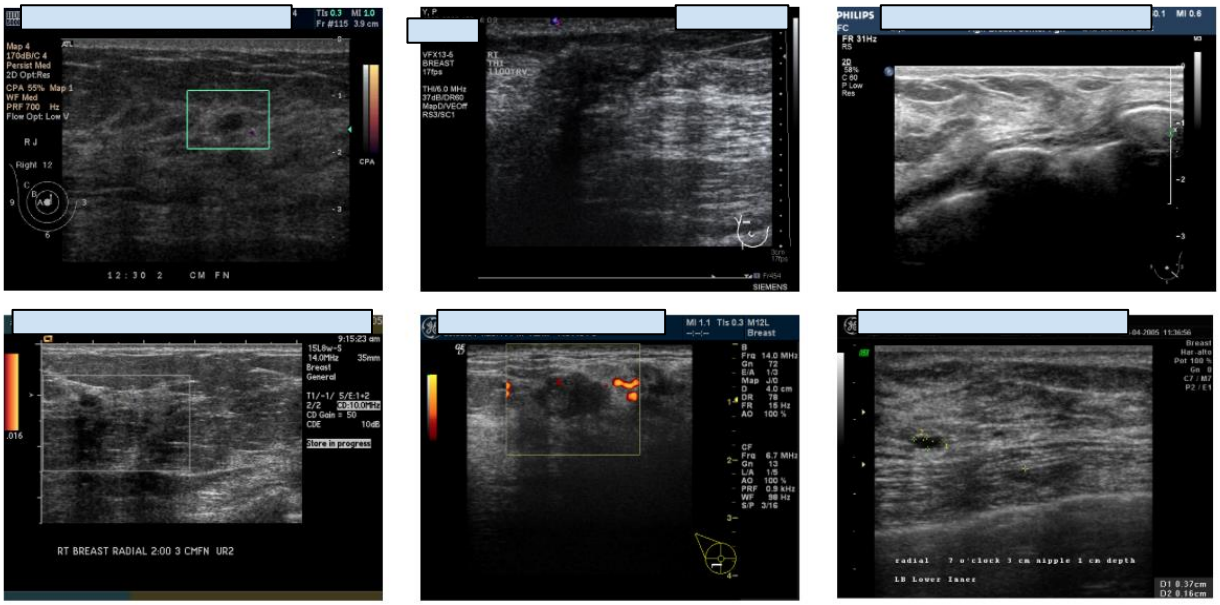


Fig 1. Ultrasound machine images. Due to privacy, all PII has been blocked out

Every ultrasound image has four main components that are important to the radiologist when they analyze the image. The location of each piece of information varies based on the manufacturer of the ultrasound machine used. First and foremost is the actual ultrasound representation of the organ that was screened. That is usually the largest part of the image and in the center. The second piece of information is the patient data, and the hospital information. This is typically at the top. The next data point is the text that

represents how the ultrasound was taken, and what the machine settings were, this is typically on the left or the right side of the image. The fourth data point is the hand typed notes that the technician created to describe the location, where on the breast, the ultrasound was taken.

3.2 Data Preprocessing

This project focuses specifically on the three channel images. Every DICOM file first needs to be examined to determine what kind of data is stored in it. If the file contains a three channel image, all metadata and the image needs to be extracted and stored separately. Any DICOM files that are missing critical data points are removed.

Once a base dataset of three channel images, with metadata, is created, we took photos of those images from a monitor in a dark environment, using a smartphone time-interval function. The digital photos of the original DICOM images then are the images used as final dataset for analysis.

All the digital photos are stored in AWS cloud with six separate file folders based on the corresponding machine type and manually checked image quality. The standard of selecting a good quality image is set as it should not have any glare spot, reflection or blur area. The overall image quality is shown in the histogram in figure 2. The final dataset only contains the images that are marked as good.

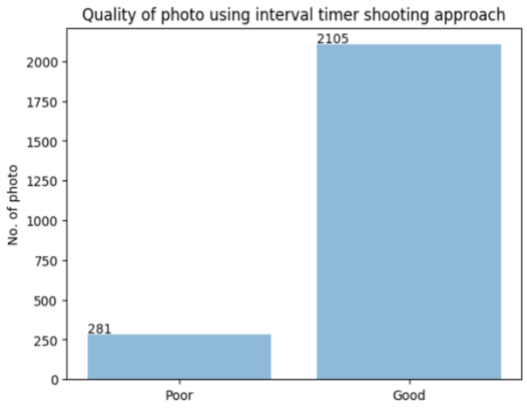


Fig 2. Number of good image quality versus poor image quality

3.2.1 Data Prep for Part 1

The dataset contains six different machine types labeled and provides 2105 good quality images. The number of instances of imaging representing each machine type are shown in figure 3., and is randomly divided into three parts, 80% of which is the training set and 10% as the test set to evaluate the machine type classification model.

We adopted deep learning-based OpenCV East, an optical character recognition (OCR) algorithm, as the text extractor to capture the text locations and then create bounding boxes on each image. With the original images (approx. 4000pi in each dimension), the size is too large in dimensions for OpenCV East to perform robust text extraction. Each image ends up being partitioned into nine clips for creating bounding boxes and stitched back into one full image. With OpenCV East, all the coordinates of bounding boxes can be generated.

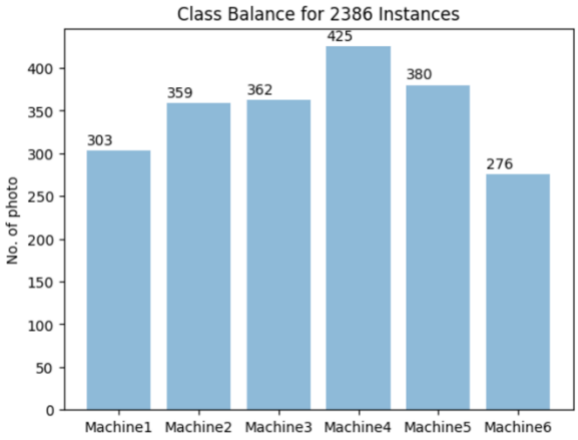


Fig 3. Number of instance of images by machine type

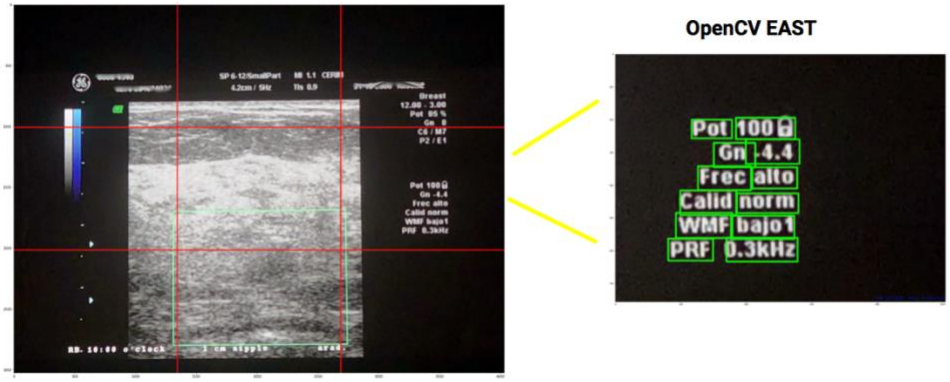


Fig 4. Example - Image Partition

In order to improve the performance of the proposed CNN model, we used data augmentation techniques to increase the image diversity by rotating, reflection, etc. By using this technique, we are able to generate an infinite number of image data for training model purpose. To unify the input image size, all the images are resized to dimension of 299x299.

3.2.2 Data Prep for Part 2

As stated above, we have all the coordinates of bounding boxes for each image. Each image has a unique quantity of geometric center as each image can have a different number of bounding boxes. At this stage, there are no attributes that can be used to define the relative coordinates. Thus, a relative coordinate calculation method is designed to overcome absolute coordinate only problems. As shown in Figure 5, the minimum distance, perpendicular distances of four edge points, who are closest to the image corners, are calculated first; and Euclidean distance of all the other points are calculated based on the four edge points. By implementing this method, each point is standardized and has 24 feature expressions.

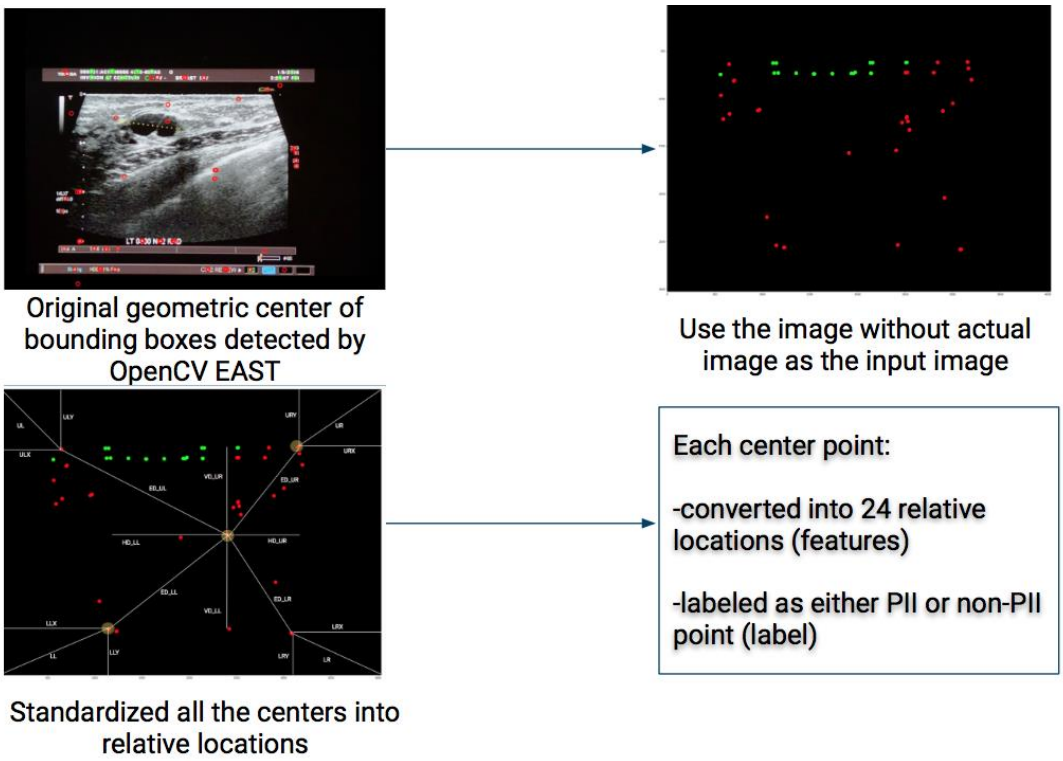


Fig 5. Feature standardization method

As a result, a single image has an average 102 geometric center points where 24 points represent PII points and 78 points represent non-PII points. To solve imbalanced data classes problem, the minority class which is PII class are randomly oversampled. A total of 155,072 points are randomly selected as training set, 10,512 points as validation set and 10,237 points as testing set. The figure 6 shows the sample of the final dataset that is used for part 2 PII prediction model.

ul	ll	ur	lr	ulx	llx	urx	lrx	uly	...	ed_lr	vd_ul	vd_ll	vd_ur	vd_lr	hd_ul	hd_ll	hd_ur	hd_lr	truth
3.848411	3434.922124	2744.277683	1227.376063	100	635	2918	2846	455	...	64394.151287	0	535	2818	2746	0	2060	61	2253	1
5.480245	3701.373394	2520.560255	1026.836891	234	346	3042	3295	733	...	124777.364686	-845	-733	1963	2216	10	1964	-17	1586	1
3.352434	3019.802146	2320.638921	767.189677	932	1158	3518	3784	764	...	167797.193099	-722	-496	1864	2130	-83	1250	-86	1451	1
0.385199	3210.618165	2348.234443	1287.158887	466	822	3125	3139	881	...	171970.765469	-2111	-1755	548	562	-70	2010	-93	1146	1
2.460533	2586.386282	2622.335028	981.099893	860	1497	3536	3173	454	...	128940.866295	191	828	2867	2504	-1067	990	-1072	1029	0
3.981629	2925.317248	2561.365456	1041.447550	651	1147	3159	3089	640	...	159130.852687	-1490	-994	1018	948	-94	1806	-118	1848	1
0.887260	3289.774612	2058.514027	623.571167	913	933	3646	3557	995	...	246871.897125	-2597	-2577	136	47	-1414	-489	-1407	211	0
5.144322	3568.398100	2753.942628	1248.032452	820	596	3542	3039	341	...	152506.759712	-1514	-1738	1208	705	10	1730	-17	1937	1
1.157651	3718.363619	2321.775398	1161.208422	495	340	3336	3353	832	...	178134.351145	-1763	-1918	1078	1095	-632	1118	-655	618	0
4.770554	3161.873653	2400.710312	924.034631	560	881	3725	3288	670	...	188984.282394	-2320	-1999	845	408	-77	2015	-104	1729	1
3.213464	3597.641728	2510.541376	725.033792	693	783	3339	3389	622	...	123151.627862	-76	14	2570	2620	-254	603	-265	1813	0
5.779934	3882.538474	2850.322262	1617.346592	228	269	3356	3373	392	...	177162.580902	-2475	-2434	653	670	127	1803	-10	1282	1
3.862084	3610.185868	2415.733843	992.480730	691	721	3125	3085	794	...	131969.294936	5	35	2439	2399	-670	121	-679	1263	0
3.774151	2654.861955	2679.018104	1167.976027	346	1418	2955	2914	587	...	84025.650179	60	1132	2669	2628	-55	1918	-71	2044	1
3.498428	3238.050803	2647.317321	1137.365816	644	819	3452	3537	519	...	143986.764312	-869	-694	1939	2024	68	2171	-10	1549	1
3.683574	2253.206160	2652.939690	1442.099164	770	1829	3724	3707	694	...	213240.663036	-2345	-1286	609	592	-323	1534	-628	602	0
7.895134	3698.095726	2522.129457	1511.600807	355	338	3153	3203	685	...	141211.681096	-1387	-1404	1411	1461	17	2182	-8	1092	1
1.755677	3675.696533	2656.711501	1078.906854	622	736	3122	3090	535	...	96318.847273	0	114	2500	2468	0	862	-7	1963	1
9.007204	2632.556932	2257.958813	1033.953577	1185	1402	3555	3006	806	...	146235.776731	-350	-133	2020	1471	-90	2012	-79	2000	1
1.261158	2806.643903	2484.091987	1173.385273	323	1233	3380	2880	636	...	163280.055861	-2284	-1374	773	273	-6	2175	-15	2159	1

Fig 6. dataset sample of center point relative locations

4.Methodology, Classification Algorithm

The first part of this project focuses on classifying each image by the manufacturer's model of the ultrasound device that was used. Since the manufacturer of the ultrasound machine determines the meaning of the text based on where it is located on the image, step one needs to be completed first. We implemented transfer learning to perform ultrasound machine type classification which uses the pre-trained Convolutional Neural Network model as the base model, and fine-tuned the model with our own photo of DICOM dataset. And then, we performed part 2 by implementing a Multiple-layer Perceptron model to classify PII and non-PII points on images.

4.1 Part 1 - Ultrasound Machine Type Classification

By using a pre-trained CNN model, we identified the manufacturer and model of each ultrasound machine that was used to create the image using the DICOM encoded metadata as the ground truth and the image as the input. In this project, we adopt InceptionV3 as the pre-trained model as it has great generalization ability and strong classification capability. InceptionV3 algorithm is based on Inception architecture with 1.2 million training images in 1,000 categories from ImageNet dataset.

With InceptionV3 as the transferable model, we are able to increase the computational efficiency. The inceptionV3 has 1,000 outputs. As shown in figure 7, we unfreeze and retrain the last 11 layers from bottom and replaced the last layer with a layer of 6 neurons as we have 6 output classes. To improve the accuracy, we remain the InceptionV3 architecture where it employs the global average pooling layer and SoftMax layer to prevent overfitting and get the probabilities of the output.

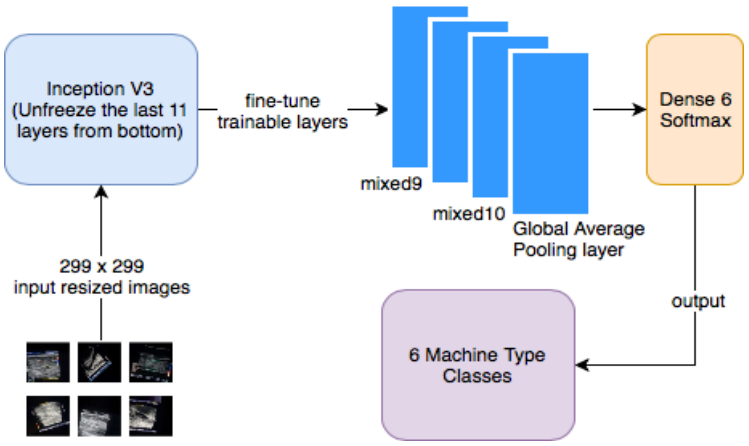


Fig 7. Part 1 framework – use InceptionV3 as pre-trained model on machine type classification

4.2 Part 2 – PII point Binary Classification

The second part of this project is to identify and extract all of the text in the photo. The following table shows the possible algorithm selection and potential challenges:

Algorithm	Challenge/Solution
Convolutional Neural Network	Both classes' locations change with different images
K-means with SVM	Number of clusters changes between machine types
Multi-layer Perceptron on all locations	Each image has variable parameters
Multi-layer Perceptron with relative locations√	Allows each location to be individually predicted

Thus, MLP with relative locations is selected as our PII classification algorithm structure where input layers are 24 PII features and machine type (25 features). As shown in figure 8, the net architecture that is found most reliable for this prediction is a three-hidden-layer 32-64-32-1 model with a total of 4,857 trainable parameters.

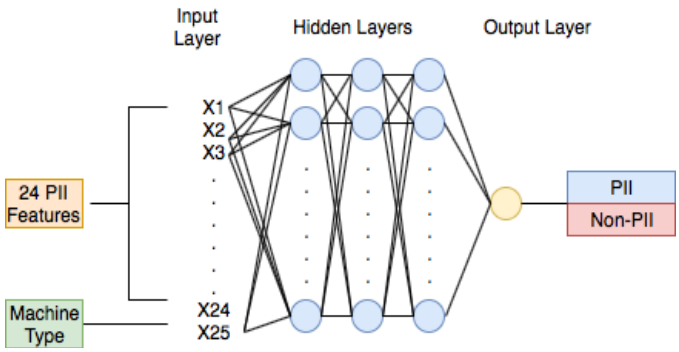


Fig 8. MLP Net Architecture

4.3 End-to-end Pipeline

The framework of proposed end-to-end application for anonymizing PII information from a photo of DICOM image is shown in figure 9. In order to obtain the efficient inputs for PII classification model, two important stages are involved – standardized text locations and predicted machine type. The final expected output of the entire project is the image of having all the PII anonymized by black bounding boxes.

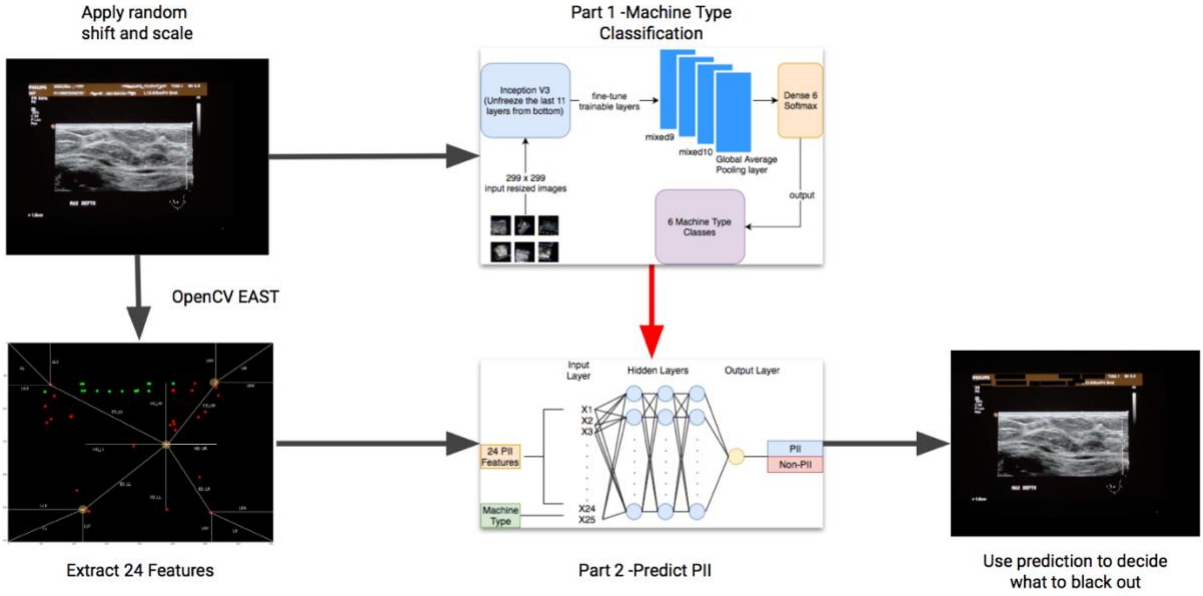


Fig 9. End-to-end framework

5. Model Evaluations and Results

5.1 Part 1 - Ultrasound Machine Type Classification

We used accuracy metric to identify which models are creating the best results. The data is separated into train, validate, and test splits. The training data is used to train the InceptionV3 CNN model, with the validation data used as a stopping criteria to avoid overfitting the model to the training data. The test data is then used to calculate the final performance of the model.

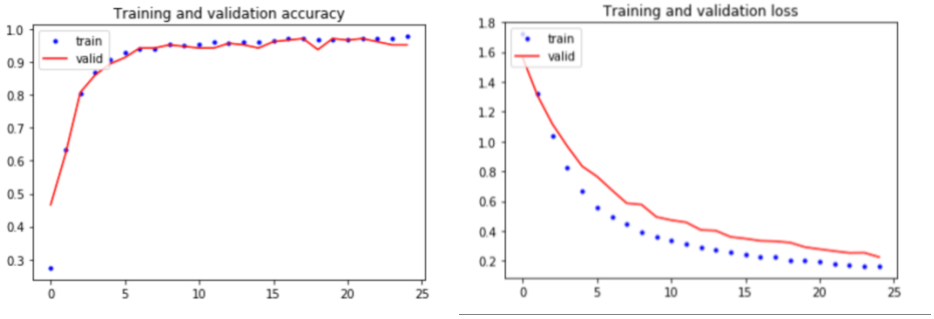
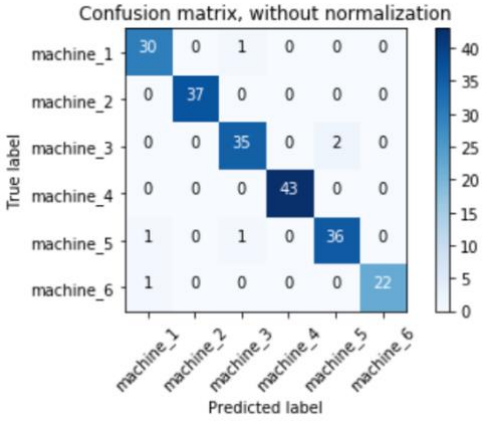


Fig 10. Machine type classification learning curve – InceptionV3 transfer learning

In this work, a total of 2105 images are used for simulating the InceptionV3 architecture. 1684 images are used for fine-tuning training and 210 images each are used for validation and testing the model. The testing result is shown as figure 11.



Machine ID	Test Set Size	Accuracy Score
1	31	0.9677
2	37	1.0000
3	37	0.9459
4	43	1.0000
5	38	0.9473
6	23	0.9565
Overall	210	0.9713

Fig 11. Machine type classification confusion matrix and table

5.2 Part 2 – PII point Binary Classification

In part 2, with 155,072 instances for training and 10,512 points for validation set and 10,237 points for testing, feature values of 25 attributes of all samples are used as input neurons and feature value of 1 attribute is used as output neuron in the classifier. Figure 12 shows the network learning curves regarding accuracy and error over epochs. Moreover, we also performed the ROC curve, shown as figure 13, to observe how well the model distinguishes the positive and negative values.

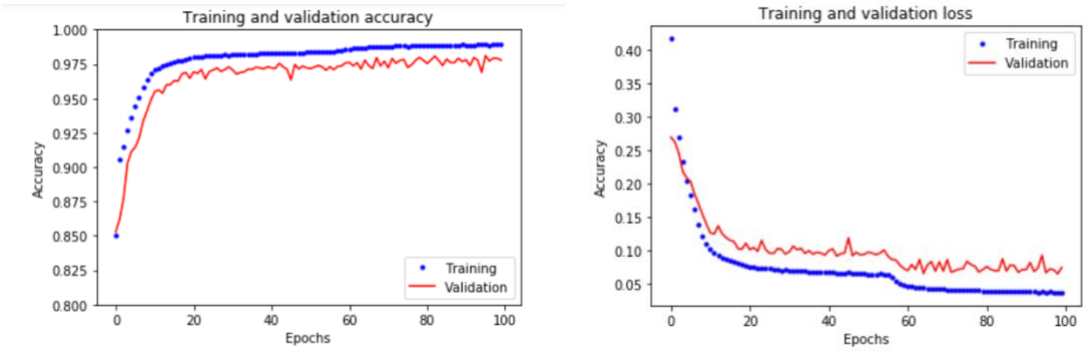


Fig 12. MLP network learning curve

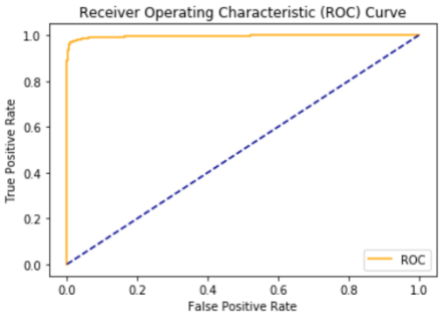


Fig 13. AUC ROC curve

The test accuracy of MLP model achieves 0.9794 and 0.9979 on ROC AUC.

5.3 End-to-end Pipeline

To testify the assumption of machine type improves PII prediction performance, we implemented three types of experiments and assessed each model shown as below:

Phase	**Model**	**Accuracy Score**
I – Factor out machine types	MLP with no machine ID parameter	0.9584
II – Factor in machine types	MLP with ground truth machine ID parameter	0.9794
III – End-to-end Model	MLP with predicted machine ID parameter	0.9740

When the machine type feature is excluded in MLP model which means only 24 PII features are included, the accuracy is lower than having machine type takes into account. Then, when it comes to end-to-end model where we used predicted machine ID from part 1, the accuracy still outperforms the model which only has PII features. Thus, we are confident on integrating the part 1 and part 2 model into an end-to-end model.

6. Conclusion

As we proposed, the final application is able to automatically anonymize PII on a photograph of DICOM image. Two efficient models using re-trained InceptionV3 CNN and cascaded neural network as MLP are implemented and suitable for the two tasks. The two models are well combined into an end-to-end model and the final accuracy reaches 0.9740. In this project, the methods which are coherent with DICOM image transmission workflow and have a great potential to benefit the teleradiology field.

7. Appendix

7.1 References

Corovic, A., Ilic, V., Duric, S., Marijan, M., & Pavkovic, B. (2018). The Real-Time Detection of Traffic Participants Using YOLO Algorithm. 2018 26th Telecommunications Forum (TELFOR), 1–4. https://doi.org/10.1109/TELFOR.2018.8611986

Coudray, N., Ocampo, P., Sakellaropoulos, T., Narula, N., Snuderl, M., Fenyo, D., & Moreira, A. (2018). Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning.(Report). Nature Medicine, 24(10), 1559–1567.

Jaderberg, M. (2015). Deep learning for text spotting. Retrieved from http://search.proquest.com/docview/1779543666/

Jin Mo Ahn, Sangsoo Kim, Kwang-Sung Ahn, Sung-Hoon Cho, Kwan Bok Lee, & Ungsoo Samuel Kim. (2018). A deep learning model for the detection of both advanced and early glaucoma using fundus photography. PLoS ONE, 13(11), e0207982.

Kim, D., & Mackinnon, T. (2018). Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. Clinical Radiology, 73(5), 439–445.

Luciani, A., Panfili, P., Furfaro, R., Ganapol, B., & Mostacci, D. (2009). Estimating water and ice content on planetary soils using neutron measurements: a neural network approach. *Radiation Effects and Defects in Solids*, *164*(5-6), 345–349.

Maruyama, T., Hayashi, N., Sato, Y., Hyuga, S., Wakayama, Y., Watanabe, H., … Ogura, T. (2018). Comparison of medical image classification accuracy among three machine learning methods. Journal of X-Ray Science and Technology, 26(6), 885–893. https://doi.org/10.3233/XST-18386

O, A., Ojo, J., & Amole, A. (2014). Application Of Fuzzy-Mlp Model To Ultrasonic Liver Image Classification. *European Scientific Journal*, *10*(12).

Pratusevich, M. (2018). Deep Learning Approximation: Zero-Shot Neural Network Speedup.

Sarangi, L., Mohanty, M., & Pattanayak, S. (2016). Design of MLP Based Model for Analysis of Patient Suffering from Influenza. *Procedia Computer Science*, *92*, 396–403.

Wei, Y., Shen, W., Zeng, D., Ye, L., & Zhang, Z. (2018). Multi-oriented text detection from natural scene images based on a CNN and pruning non-adjacent graph edges. Signal Processing: Image Communication, 64, 89–98. https://doi.org/10.1016/j.image.2018.02.016

Youbao Tang,& Xiangqian Wu. (2017). Scene Text Detection and Segmentation Based on Cascaded Convolution Neural Networks. IEEE Transactions on Image Processing, 26(3), 1509–1520. https://doi.org/10.1109/TIP.2017.2656474

7.2 Work Log

Individual Work – Robert Warfsman
Date, Timespan, Hours, Description
09/04, 4pm - 6pm - 2hr, Discussion with SonaVista on project goals
09/05, 6pm - 8pm - 2hr, Initial data gathering from SonaVista
09/08, 6pm - 9pm - 3hr, Data exploration
09/10, 6pm - 8pm - 2hr, Creation of AWS instances for data analysis
09/11, 7pm - 9pm - 2hr, Loaded data into the AWS instance and prepared the instance
09/13, 5pm – 8pm - 3hr, group meeting to discuss the objectives of the project
09/16, 8pm – 10pm, 2hr, Initial data analysis
09/18, 6pm – 7:30 pm, 1.5hr, literature review
09/20, 9pm – 10pm, 1hr, literature review
09/24, 1pm – 4pm, 3hr, proposal writing and research
09/25, 10am - 1pm, 3hr, presentation work
09/26, 8pm - 10pm, 2hr, presentation work
09/27, 7pm – 8:30pm - 1.5hr, group meeting to discuss project details and review proposal
09/30, 4pm - 6pm - 2hr, meeting with SonaVista
10/2, 6pm - 8pm - 2hr, proposal and presentation work
10/4, 9:30am - 11am - 1.5hr, presentation and proposal work
10/4, 1:30am - 4pm - 2.5hr, presentation and proposal work
10/04, 6pm – 7:30pm, 1.5hr, group meeting to discuss project and prepare the presentation
10/6, 9am -12pm -3 hrs, clean data and take photos for additional tests
10/11, 5pm -- 6pm, 1 hr, group meeting to discuss the project improvement
10/10, 8 - 10, 2hrs, work on photos
10/12, 5 - 7, 2hrs, work on presentation rework
10/14, 10 - 3, 5hrs, work on photos
10/15, 6:30 - 10:00, 3.5hrs, work on presentation
10/16, 10:00 - 4:00, 6hrs, tested photo images with transfer learning using InceptionResNetv3
10/17, 8:00 - 10:00, 2hrs, worked on CNN
10/20, 7:00 - 9:00, 2hrs, data point extraction
10/21,7:00 - 9:00, 2hrs, data point extraction
10/25, 7pm - 9pm, 2hr, group meeting - discussion on next steps
10/27, 7pm – 8:30pm, 1.5hr, group meeting to discuss project details and model selection
11/2, 9:00 - 1:00, 4hrs, data point extraction and testing pixel data conversion
11/3, 10:30 - 2:00, 3hrs, extract 12 data points from each image for MLP
11/06, 2pm – 5:30pm, 3.5hr, group meeting to discuss project and prepare the presentation
11/7, 8:30 - 10:30, 2hrs work on MLP for text prediction
11/9, 10:00 - 5:00, 7hrs created new extraction for 24 points of data
11/10, 12:00 - 6:00, 6hrs updated the MLP for 24 features, grid search for optimal MLP configuration
11/16, 4:00 - 6:30, 1.5,,worked on next update presentation
11/17, 8pm -- 9pm, 1 hr, group discussion on end-to-end integration process
11/21, 9:00 - 10:30, 1.5 worked on end-to-end integration for the final test
11/24, 8:30 - 11:00, 2.5 finished end-to-end integration
12/2, 8:00 - 10:00, 2hrs final presentation work
12/3, 7:30 - 10:00, 2hrs final presentation work
12/4, 4:00 - 5:00, 1hr, final presentation and pitch practice
12/13, 9:30 - 11:00, 1.5, cleaned up code files for submission
12/14, 10:00 - 1:00, 3hrs, finished cleaning up code files for submission
12/15, 10:30 - 12:00, 1.5 hrs, worked on final report
Total - 112 hours

Individual Work – Wanfei Luo
Date, Timespan, Hours, Description

09/13, 5pm – 8pm, 3hr, group meeting to discuss the objectives of the project
09/17, 2pm – 5 pm, 3hr, learning image segmentation-related knowledge
09/19, 7pm – 9pm, 2hr, research and literature review
09/27, 9pm – 12pm, 3hr, proposal writing and literature review
09/27, 1pm – 5pm, 4hr, proposal writing and research
09/27, 7pm – 8:30pm, 1.5hr, group meeting to discuss project details and review proposal
10/01, 1pm – 5pm, 5hr, literature review and ML self-study
10/02, 4pm – 7pm, 3hr, proposal and slide finalize
10/04, 10am – pm, 2 hr, presentation preparation
10/04, 6pm – 7:30pm, 1.5hr, group meeting to discuss project and prepare the presentation
10/11, 5pm -- 6pm, 1 hr, group meeting to discuss the project improvement
10/13, 5pm -6:30pm, 1.5 hr, research on text detection of medical monitor and relevant prior works
10/15, 10am -- 12 pm, 2 hr, research on text detection of medical monitor and relevant prior works
10/15, 1:30pm -- 5:30pm, 4hr, prepare for presentation 2
10/17, 8pm -- 9pm, 1 hr, photo quality check
10/18, 10:30am -- 1pm, 2.5hr, photo quality check
10/21, 9:30pm --10:30pm, 1hr, data preparation and research
10/22, 10am -- 12:30pm, 2.5hr, project update and slides editing
10/22, 1pm -- 4pm, 3hr, research on CNN and slides editing
10/23, 3:30pm -- 5:30pm, 2hr, prepare for a presentation and research on AWS use case (48.5)
10/25, 7pm - 9pm, 2hr, group meeting - discussion on next step based on presentation 2 feedback
10/27, 9pm – 12pm, 3hr, image preprocessing - OpenCV East
10/27, 1pm – 5pm, 4hr, Transfer learning research (InceptionV1,2,3,4 and VGG16) and coding
10/27, 7pm – 8:30pm, 1.5hr, group meeting to discuss project details and model selection
11/01, 1pm – 5pm, 5hr, literature review and CNN and InceptionV3 set-up
11/02, 4pm – 7pm, 3hr, read and learn teammate's code
11/04, 10am – pm, 2 hr, presentation preparation - flow chart design and edit
11/06, 2pm – 5:30pm, 3.5hr, group meeting to discuss project and prepare the presentation
11/11, 5pm -- 6pm, 1 hr, MLP research and work
11/13, 5pm -6:30pm, 1.5 hr, MLP algorithm programming review and study
11/15, 10am -- 12 pm, 2 hr, bounding box adjustment work
11/15, 1:30pm -- 5:30pm, 4hr, prepare for presentation 3
11/17, 8pm -- 9pm, 1 hr, group discussion on end-to-end integration process
11/18, 10:30am -- 1pm, 2.5hr, make poster and final presentation slides
11/22, 10am -- 12:30pm, 2.5hr, project update and slides editing
11/24, 3:30pm -- 5:30pm, 2hr, prepare updates and finalize poster
12/3, 12pm-5pm, 5hr, final slide editing and 10-min presentation prep
12/11, 10am – 5pm, 7 hr, work on final report
12/12, 11am – 5pm, 6 hr, work on final report
12/14, 9pm – 12am, 3hr, finalize the final report submission

Total – 121 hours