Comparison of Attention Methods for Mammogram Classification

Shawn Ray
Department of Computer
Science & Engineering
University of Nevada, Reno
Email: shawnray@nevada.unr.edu

I. Abstract

Attention-based neural networks have been shown to improve classification and segmentation performance for cancer detection in mammograms. Of the many attention mechanisms that have been used for cancer detection in mammograms, the Cross-view Attention Module (CvAM) is one of the most promising. Cross-view Attention is a multi-view attention method developed to retain relevant information between each of the four views of a mammography exam. Although CvAM was shown to improve upon the baselines of ResNet50 and ResNet50 with the Convolutional Bottleneck Attention Module (CBAM), little effort has been made to compare CvAM with state of the art attention methods used for breast cancer classification. In this paper we present a comparison of CvAM with CBAM, Squeeze and Excitation (SE), and Multi-Head Attention (MHA). We implemented each attention module within ResNet50, DenseNet201 and EfficientNetB4 to determine which attention module and base network combination performs best.

II. Introduction

Mammography is a technique used to locate and diagnose potential tumors located in the breasts of humans. This technique involves using X-rays to create images of the internal structure of the breasts. This is the most common form of diagnostic tool employed by Oncologists when concerns regarding breasts come into topic. Traditional Mammograms can be taken from different views. The Bilateral Craniocaudal is taken from the top of the breast where Mediolateral Oblique is taken from the side.

A preferred mammography exam will consist of at least four images. The images will be taken of both the left and right breast, and the two aforementioned views. The four images would be a right Mediolateral Oblique, left Mediolateral Oblique, right Bilateral Craniocaudal, and left Bilateral Craniocaudal; referred to as R-MLO, L-MLO, R-CC, and L-CC respectively. Cancer is most likely to only occur in one breast at a time. Despite this, each image taken contains relevant information regarding diagnosis. In clinical practices, a radiologist will examine each of the four mammograms for regions of interest such as masses, micro-calcifications or asymmetries. If they find an abnormality in one image, the radiologist can focus their attention on the expected location, shape, size and brightness of the mass to determine if the abnormality is also present in the other images.

In an attempt to replicate the procedures of a radiologist, Zhao et al., 2020, developed the Cross View Attention Module (CvAM) [XZW20]. The Cross-View Attention Module (CvAM) combines information from both projections (CC, MLO) through channel attention, and from both breasts (left, right) through spatial attention. CvAM has been shown to outperform the ResNet50 and ResNet50 + CBAM baselines for the DDSM dataset. However, there are no comparisons between CvAM and state of the art attention methods, so it is difficult to conclude if CvAM is still a state of the art method. Furthermore, the authors of the CvAM paper only tested CvAM within a ResNet50 framework, so it is possible that higher scores could be achieved if CvAM were places within a different network backbone. Finally, CvAM can be used to turn single-view networks into multi-view networks, but the utility of this function has not been fully explored. In this paper, we plan to test the performance of CvAM, CBAM, SE and MHA within ResNet50, DenseNet201 and EfficientNetB4 backbone networks. Thus, our paper will attempt to address the following questions:

*1) **Is CvAM still one of the best attention methods for breast cancer classification?**:*

*2) **What network backbone + attention module combination provides the best scores?**:*

*3) **Can we use CvAM to effectively develop multi-view networks in conjunction with single-view attention methods?**:*

III. Review of Methods

A. *Convolutional Block Attention Module (CBAM)*

The baseline attention method for breast cancer classification in mammograms is the Convolutional Bottleneck Attention Module (CBAM). The Convolutional Block Attention Module (CBAM) is a method of attention in feed-forward convolutional networks. Given an intermediate feature map, the CBAM module infers attention maps for both the spatial and channel attention. These channel and spatial maps are generated in their respective modules. One for the spatial attention and one for the channel attention. These attention maps are then multiplied to the input feature map for adaptive feature refinement. The benefit of the CBAM module is the ability to supplant CBAM into any feed-forward network to benefit that network with spatial and channel attention.

B. *Cross-view Attention Module (CvAM)*

The Cross-view attention module (CvAM) as detailed by [XZW20] can potentially be a valuable asset in the mammography field. The use of attention modules between the layers allows the model to retain relevant information between the four views. The algorithm functions by forming two modules referred to as the *Bi-lateral attention module* and the *Bi-projection attention module*. The Bi-lateral attention module calculates a feature map between the left and right breast, where the Bi-projection attention module calculates a spatial vector for the different views.

The CvAM algorithm takes the intermediate feature maps for each of L-CC, L-MLO, R-CC, and R-MLO respectively. CvAM is a combination of the Bi-lateral attention module which calculates 2-D spatial attention maps between the left and right breast for the two views CC and MLO; and the Bi-projection attention module which calculates a 1-D channel attention between the two views CC and MLO for the left and right breast.

1) ***Bi-lateral attention module****:* The spatial attention map is generated by combining the left and right feature maps of the breasts. It does this quite simply because the images for both breasts are relatively spatially aligned with just one image being horizontally flipped. The generation of the spatial attention map is done by first applying average-pooling and max-pooling along the channel axis of feature maps for both the left and right breast. These are then separately concatenated into a feature descriptor. Applying pooling regions along the channel axis highlights informative regions. Passing through a convolution layer, a ReLU layer, another convolution layer and finally passing through a sigmoid layer for form the spatial attention map.

2) ***Bi-projection attention module****:* The Bi-projection attention module attempts to aggregate information in both the views of the same breast. That is, the CC and the MLO view from the same breast. It is often difficult to acquire spatial information between images acquired from different views. Knowing this, the spatial dimension is squeezed down into a feature vector through average-pooling and max-pooling. This vector is then forwarded through a multi-layer perceptron with one hidden layer to form the channel attention map.

C. *Squeeze and Excitation Module (SE)*

Squeeze and Excitation (SE) is a technique used in neural networks to improve the performance of a model by adaptively re-calibrating the feature maps of a convolutional neural network (CNN). The goal of SE is to improve the representational power of the network by using global information to selectively excite the important feature maps and squeeze the less important ones.

The technique of SE consists of two main steps:

- **Squeeze:** The feature maps of a CNN are first passed through a global average pooling layer, which reduces the spatial dimensions of the feature maps to a single vector of values representing the global information of the feature maps.
- **Excitation:** The global information is then used to adaptively re-calibrate the feature maps by applying a weighting factor to each feature map. The weighting factor is learned through a fully connected (FC) layer, which takes the global information as input and outputs a weighting factor for each feature map. The weighting factors are then used to scale the feature maps, effectively emphasizing or suppressing the importance of each feature map.

D. *Multi-Head Attention (MHA)*

A Well known method where the attention method is run multiple times in parallel. This method serves as the basis of the transformer model described by [VSP+17]. The transformer model is formed by stacking an encoder and decoder architecture. The encoder employs a residual connect around the two sub-layers followed by a layer normalization. The Decoder uses the two sub-layers in each encoder step and adds on a third additional layer and additionally performs multi-head attention over the output of the encoder stack. An attention fucntion can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors as described [VSP+17]. This means the output is computed as a weighted sum of the values, where the weight is computed as a compatibility of each query key pair. So this transformer forms its

attention a normal way, but Multi-Head Attention (MHA) is the attention ran multiple times in parallel.

IV. Model Development

A. General

For all models, we used a relatively similar tail section. The last layer of the baseline models is always the average pooling layer, which feeds into a dropout layer, then a dense layer with 256 nodes with relu activation, then a batch normalization layer. This block is repeated two more times, then a final dense layer is used with 3 nodes to output the categorical classification results. The amount of dropout and number of nodes in each layer was changed to avoid overfitting in some cases. For the backbone models without attention, we surprisingly found that the model produced higher scores when all layers were trainable. Thus, all models had all layers trained.

B. ResNet50

For the baseline scores, ResNet50 was imported via the Keras applications module. Imagenet weights and average pooling were used. For ResNet50 models with attention modules, we used the full source code for ResNet50 written in Keras. For attention modules that don't reshape the input (such as CBAM and CvAM), we inserted the modules into ResNet50 after the 3rd, 4th, and 5th convolutional blocks, since this was the setup proposed in the original CvAM paper. For attention modules that do reshape the input, such as MHA, we placed the attention module after the 5th convolutional block, and ensured that the output of the attention module would be batch size x 2048.

C. DenseNet201

For the baseline scores, DenseNet201 was imported via the Keras applications module. Imagenet weights and average pooling were used. For DenseNet models with attention that didn't reshape the inputs, we placed the attention modules after the second and third transition blocks. For MHA, we placed the module after the last DenseNet layer before the global average pooling layer.

D. EfficientNetB4

For the baseline scores, EfficientNetB4 was imported via the Keras applications module. Imagenet weights and average pooling were used. For EfficientNet models with attention that didn't reshape the input, we placed the modules at the end of each MBConvolution block (7 in total). For MHA, we placed the module after the last DenseNet layer before the global average pooling layer.

E. CvAM

CvAM was implemented by changing CBAM to calculate the channel and spatial attention for each view, and multiplying the refined feature maps together in parallel as shown in the CvAM paper. In order to make CvAM multi-view, we increased the dimensions of the input tensor. Instead of a normal 4D input tensor for images [batch size, width, height, channels], the input for the CvAM model is a 5D tensor [batch size, view type, width, height, channels]. Due to computational constraints, the largest possible batch size was 32. However, because the number of samples in the batch is actually multiplied by 4 (for each view type), we ended up with 128 images per batch. After reading each batch of data, the 5D tensor is split into 4 4D tensors (one for each view). The previous algorithm developed by the original authors of CvAM is given by Figure 1. 3.

Algorithm 1 CvAM

Input: Intermediate feature maps of 4 views in a screening mammography exam, $\{\mathbf{F}_L^{CC}, \mathbf{F}_L^{MLO}, \mathbf{F}_R^{CC}, \mathbf{F}_R^{MLO}\} \in \mathbb{R}^{c \times h \times w}$.
for $p = CC, MLO$ **do**
  Calculate bi-lateral attention $\mathbf{A}_p \in \mathbb{R}^{1 \times h \times w}$ for projection p based on $\mathbf{F}_p^L$ and $\mathbf{F}_p^R$ according to Equation 1.
end for
for $s = L, R$ **do**
  Calculate bi-projection attention $\mathbf{A}^s \in \mathbb{R}^{c \times 1 \times 1}$ for breast s based on $\mathbf{F}_{CC}^s$ and $\mathbf{F}_{MLO}^s$ according to Equation 2.
end for
for $p = CC, MLO$ **do**
  for $s = L, R$ **do**
    Calculate the refined feature map $\mathbf{F}_p^{s*}$ for side s and projection p according to Equation 3.
  end for
end for
Return: Refined feature maps $\mathbf{F}_L^{CC*}, \mathbf{F}_L^{MLO*}, \mathbf{F}_R^{CC*}, \mathbf{F}_R^{MLO*}$.

Figure 1. The Cross-view attention Module algorithm

[XZW20] mentioned the intermediate feature maps of the four views make independent forward passes prior to being fed to either the Bi-lateral attention module or the Bi-projection attention module. Our solution is slightly different than the original algorithm developed by the authors of CvAM. After our data is split into four 4D tensors (one for each view), we pass each view through independent branches of the baseline model. Similar to the original CvAM paper, we include a CvAM block after the 3rd, 4th and 5th convolutional blocks. The four inputs are then separately fed into their own respective CvAM layers, where cross-view attention is computed for the relevant views. The feature maps from each view are then concatenated at the end of the baseline model, and this concatenated feature map is fed into our normal tail section with dense, dropout and batch normalization layers. Learning rates, model tail architecture, and image preprocessing weren't changed between multi-view and single view experiments.

F. *Multi Image and Single Image Tests*

After much trial and error we got the multiple image generation and concatenation scheme for CvAM to work, and the model does train, but not nearly as well as the single image generation models. Furthermore, the fault is not necessarily with CvAM, but instead is with the multiple image generation and concatenation pipeline, which we can see from the fact that even the baseline models perform poorly in the multiple image generation pipeline. Thus, in order to understand the performance of CvAM, we acquired two sets of results. The first set of results is from single image generation, so these results don't include CvAM, but they are the best possible scores we have from each individual model besides CvAM. The second set of results is from the multiple image generation. Although these results are all much worse than the results from the single image generation, they act as a comparison point between CvAM and the other baseline and attention models.

G. *CBAM*

Since CBAM was developed, it has been widely adopted as a useful attention method in neural networks. Many versions of CBAM exist, and we have chosen to use a version written in Tensorflow. In order to implement CBAM within a backbone network, we simply insert a CBAM layer after the last three convolutional blocks for ResNet50. Our version of CBAM had a kernal size of 7 and used a ratio of 8.

H. *Squeeze and Excitation (SE)*

Our squeeze and excitation block used a ratio of 8 and was developed in a standard way according to the literature. Similar to CBAM, the SE block was also placed in between the last three convolutional blocks for ResNet50, after each group for EfficientNetB4, and after the 3rd and 4th convolutional blocks of DenseNet201

I. *Multi-Head Attention (MHA)*

Our Multi-head attention module is similar to the standard multi-head attention developed by Vaswani et al., 2017. We tried head values of 4,8 and 16, and weight dimensions of 32, 64 and 128, and found that the head value of 8 and weight dimension of 64 gave the best performance. The output of multi-head attention is a 2D feature map, so it can't be placed in between convolutional blocks. Instead, we placed the multi-head attention block at the end of the baseline model, and didn't use average pooling.

V. Data and Preprocessing

A. *Mini-DDSM*

The dataset we have used is the **Mini-DDSM** dataset which contains 602 Normal examinations. 679 Cancerous examinations and 671 Benign examinations. All 3 of these subsets contained the CC and MLO view for both the left and right breast. We treid a variety of image preprocessing methods such as cropping the borders, horizontal flipping, global thresholding and applying CLAHE for contrast augmentation. However, we found that all models trained best without our preprocessing methods. Most likely, there is an error in our preprocessing that results in lower scores, but we can't find it. Thus, the only preprocessing was through the rescale option built into ImageDataGenerator. All images were rescaled to have a minimum of 0 and maximum of 1. All images were resized to 400x400 pixels. The dataset followed a 70-15-15 split of training, validation, and testing sets respectively.

VI. Training/Testing

A. *General*

Our data pipeline used the Keras function flow-from-dataframe to get the correct images for each batch. The image data was resized to 400x200 (except for EfficientNet, where computational constraints demanded a size of 320x160), and the only preprocessing step was to normalize the data to give the pixel values a minimum of 0 and maximum of 1. For all experiments we used a cyclic learning rate. Generally the max learning rate was around 1e-3 and the initial learning rate was around 1e-7. If the model was overfitting we lowered the learning rates, and if the model wasn't training enough we raised the learning rates. We trained for up to 200 epochs, and stopped early if the validation loss didn't decrease in 20 epochs. The lowest validation loss weights were then restored for the final model, and the test scores were generated through the Keras model.evaluate() method (called on the testing image data generator).

Training was done on a GTX 1080, RTX 3070, Google Colab and Kaggle. We predicted categorical classification (benign, cancerous, and normal). We used a standard categorical crossentropy loss with Adam to train and test the models. For all models except CvAM, we used all the images for training, validation and testing (70-15-15 split). For CvAM, we only used images where each of the four views from a single exam were present. From a theoretical standpoint, the shuffling feature in ImageDataGenerator should be set to "False" for CvAM, since the order of image generation matters. However, we were not able to train the model in this way, as shuffling leads to robust generalization that is needed for decent model performance.

VII. Results and Discussion

A. *Single Image Results*

1) ResNet50 == RN:

Metric	RN	RN+CBAM	RN+CvAM	RN+MHA	RN+SE
Accuracy	.783	.804	N/A	.63	.79
AUC	.923	.933	N/A	.84	.934

2) EfficientNetB4 == EB4:

Metric	EB4	EB4+CBAM	EB4+CvAM	EB4+MHA	EB4+SE
Accuracy	.816	.84	N/A	.838	.841
AUC	.94	.957	N/A	.955	.954

3) DenseNet201 == DN:

Metric	DN	DN+CBAM	DN+CvAM	DN+MHA	DN+SE
Accuracy	.829	.814	N/A	.658	.828
AUC	.952	.945	N/A	.868	.947

B. Multi Image Results

1) ResNet50 == RN:

Metric	RN	RN+CBAM	RN+CvAM	RN+SE
Accuracy	.612	.628	.611	.543
AUC	.824	.829	817	.761

2) DenseNet201 == DN:

Metric	DN	DN+CBAM	DN+CvAM	DN+SE
Accuracy	.49	.357	.395	.396
AUC	.728	.52	.591	.573

C. Discussion

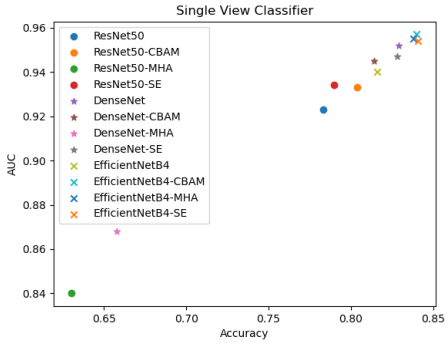

Figure 2. Single view results accuracy and AUC. ResNet50 backbones are shown in circles, DenseNet as stars and EfficientNet is shown as an X.

We set out to answer three questions with this project: is CvAM still one of the best attention methods? What is the best combination of common network backbones with common attention modules? Can we use CvAM to develop multi-view methods

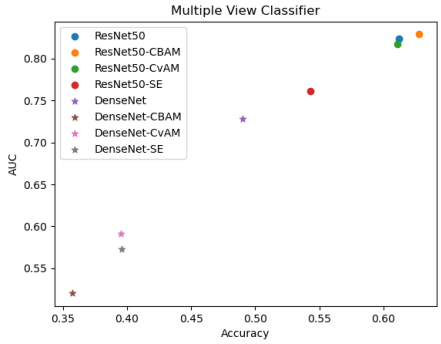

Figure 3. Multi view results accuracy and AUC. ResNet50 backbones are shown in circles, DenseNet is shown as stars.

that combine CvAM with other attention modules? We were not able build any models that attempt to answer the 3rd question due to time constraints, but we do have significant results that allows us to answer the first two questions.

First off, CvAM doesn't appear to be one of the best attention methods. Although we were not able to implement CvAM exactly as the original authors described, we tried to follow the paper as closely as possible in our model construction. Also, we were not able to get our multi-image generator to work well for any model, so it is possible that the models weren't being trained correctly, and thus the comparison is inaccurate. However, the results of the multi-image generator do indicate that CvAM is worse than the baseline model, and none of the CvAM results even come close to any of the single image results. Thus, according to our testing, CvAM is not the best attention module for breast cancer detection in mammograms.

Second, we find that the best model combination is EfficientNetB4+CBAM. In general, both CBAM and SE perform better than the baseline model and MHA. MHA works well with EfficientNetB4, which indicates that the poor performance in other backbones may be a result of poor module placement within the backbone. Oddly enough, the highest baseline performance is for the DenseNet201 backbone, but DenseNet201 with the attention modules ends up performing worse than the baseline.

REFERENCES

[CKL+18] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. *Advances in neural information processing systems*, 31, 2018.

[DGPPK22] Rosimeire A. Roela Gabriel Vansuita Maria Aparecida Azevedo Koike Folgueira Daniel G. P. Petrini, Carlos Shimizu and Hae Yong Kim. Breast cancer diagnosis in two-view mammography using end-to-end trained efficientnet-based convolutional network. *IEEE*, 10, 2022.

[ELC22] Martina Valleriani Eleonora Lopez, Eleonora Grassucci and Danilo Comminiello. Multi-view breast cancer classification via hypercomplex neural networks. *arXiv*, 14, 2022.

[GLMH21] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks, 2021.

[JM19] Xiang Li Hongwei Li Bjoern H. Menze Rongguo Zhang Wei-Shi Zheng Jiechao Ma, Sen Liang. Cross-view relation networks for mammogram mass detection. *arXiv*, 2019.

[KJGC17] Yiqiu Shen Nan Wu S. Gene Kim Eric Kim-Laura Heacock Ujas Parikh Linda Moy Krzysztof J. Geras, Stacey Wolfson and Kyunghyun Cho. High-resolution breast cancer screening with multi-view deep convolutional neural networks. *arXiv*, 2017.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[WPLK18] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. *CoRR*, abs/1807.06521, 2018.

[WYC+21] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. *arXiv preprint arXiv:2108.00154*, 2021.

[XZW20] Luyang Yu Xuran Zhao and Xun Wang. Cross-view attention network for breast cancer screening from multi-view mammograms. *IEEE*, 10:1050–1054, 2020.