Lightweight Mask RCNN for Warship Detection and Segmentation

JINYOUNG PARK AND HOSEOK MOON
Department of Military Science, Korea National Defense University, Nonsan 10544, South Korea
Corresponding author: Hoseok Moon (hsmoon0329@kndu.ac.kr)

ABSTRACT As the term X(Everything)+AI indicates, AI is applied in every aspect of current societies. Likewise, the military requirements for AI are increasing as well. AIs that automatically detect and classify objects are required for surveillance and reconnaissance. Especially in terms of naval operation, identifying types of warships and recognizing mounted armaments have significance as the first step of the operation. This study is the proposal of an AI model that can identify warships’ type and weapon by analyzing video information taken on sea, and evaluate threat priority and response level. The proposed model is based on Mask RCNN, the Image Segmentation model, but was designed in a lightweight version, so that it could be used on a platform of the vessel where the use of high performing computers is limited. To lightweight the model, the former backbone was replaced with MobileNetV2, and the convolution operation of the RPN was replaced with Depthwise Separable Convolution operation, which operates respectively in each channel. The lightweight Mask RCNN model showed 64% lower number of parameters compared to the base model. However, its mAP, the classification accuracy, was similar with the base model.

INDEX TERMS Warship detection and segmentation, lightweight deep learning, Mask RCNN, MobileNet.

I. INTRODUCTION
The operation process at sea is a continuation of “see(search & detection)-E-value-Action”. For the action, the final stage, the process of quickly and accurately evaluating what has been detected is crucial. Especially in case of detecting a large number of unknown warships during naval patrol operation, prioritizing threats of warships is important in preparing for the efficient counterattack. The priority of threat may vary depending on the class of warships and the weapons they possess. For instance, a battleship with long-range guided missiles and aircraft carriers should be dealt with a higher level of interest because of its high threat, whereas combat support ships such as auxiliary ships can be dealt with a relatively low level of interest. Also, even on the same battleship, a ship with guided missiles can be a bigger threat than a ship with only guns. The final classification process of the object detection is usually done by human eyes, though a warship has an optical camera and radar for detection. However, the process of human monitoring has limitations because human eyes cannot observe objects perfectly for a long time, and a capability of classification varies depending on an individual. Therefore, in order to reduce the fatigue of eye monitoring and avoid errors caused by it, the process of warship detection and classification needs to be automated. One way to do this is to apply an artificial intelligence model to the collected video data of warships on sea. Convolution Neural Network (CNN) [1] can extract and classify features of various warship images, and thereby enables automatic detection and classification. The performance of the CNN model has been proven in an image recognition contest ILSVRC(ImageNet Large-Scale Visual Recognition Challenge), as it has shown since 2015 an error rate 5% lower than that of human recognition [2]. The examples of object detecting and classifying algorithm models based on CNN include Fast RCNN [3], Faster RCNN [4], Mask RCNN [5], SSD [6], YOLO [7], [8] [9], [10], YOLACT [11]. Among them, Mask RCNN has been used and was verified of its operation in various fields including agriculture [12], medical [13], [14], and human motion recognition [15], [16]. In this study, we used the Mask RCNN algorithm to develop a model that detects warships, identifies mounted weapons, and determines the priority of response. However, though Mask RCNN has a high accuracy rate in classification, it requires numbers of parameters such as ResNet [17], [18], which works as a backbone in the process of extracting the Features

The associate editor coordinating the review of this manuscript and approving it for publication was Ashish Mahajan ©.
from video. Due to its shortcoming in requiring a substantial amount of calculations in learning and analyzing, the training process of Mask RCNN requires high-performance Computing Resources such as Computer Process Unit (CPU), Graphic Process Unit (GPU), and memory. Otherwise Mask RCNN will not operate properly. In this study, we used an artificial intelligence model based on Mask RCNN that can classify classes of warships, identify weapons, and assist in determining priority of response. We proposed a lightweight version of Mask RCNN model that can operate on platforms where high performance computers cannot be mounted, such as warships, helicopters, drones, unmanned surface vehicles (USV), and unmanned air vehicles (UAV). The proposed Mask RCNN has a reduced number of parameters, but the performance of the model is similar to the original version. In order to reduce the weight of the model, we changed a backbone, which deals with the process of extracting features from the inputted image, to MobileNet V2 [19]. Also, the proposed model uses Depthwise Separable Convolution in operation, which proceeds the convolution operation of the Region Proposal Network (RPN), which is the network that suggests an interest area in Feature Map, separately in each channel. The contents of this study is as follows. Chapter 2 reviews the existing works which use deep learning for the maritime object detection. Also, in this chapter, the main methodologies of this study Mask RCNN and deep learning lightening techniques are explained. Chapter 3 explains the data and model construction process as a research method. Chapter 4 evaluates the performance of the configured model and presents a way to utilize it. Chapter 5 is a conclusion and a summary of the research, and the proposal for the future research directions.

II. RELATED WORK
A. MARITIME OBJECT DETECTION
Research on maritime object detection has been usually conducted for navigation safety, autonomous driving systems, and military use. References [20]–[23] applied Mask RCNN to detect merchant ships and marine buoys on sea. References [24]–[26] detected warships by applying Mask RCNN and YOLO. However, [24]–[26] has its limitation in being only capable of detecting warships, and not being able to classify the class of warship nor distinguish between mounted guided missiles and guns. To reduce the weight of Mask RCNN, the base backbone was replaced with MobileNet V1 [27] and the number of convolutional computation kernels of the head was reduced in [28]. Through this process the size of the model was reduced from 245Mb to 47.1Mb. However, mAP, the classification accuracy of the model, was decreased from 0.57 to 0.36. In case of [29], the model size was reduced from 170Mb to 84Mb, as the spatial convolution operation of Mask RCNN was replaced with Depthwise Separable Convolution. However, as in the case of [28], the mAP decreased from 0.87 to 0.78, showing that reducing weight will cause degradation in performance. The existing studies showed their limitation in not being able to promote detailed observation of warships as they simply focused on detecting them, and the lightweight RCNN had a problem in accuracy.

B. MASK RCNN
Mask RCNN was created by adding Mask Branch to the RPN of Faster RCNN, and is capable of image classification, localization, and segmentation. As shown in Figure 1, the model is divided into two stages. Stage-1 extracts features from the input image and gives proposals of regions of interest. Stage-2 handles Bounding-Box, Classification, and Masking processes with the extracted regions of interest. Stage-1 receives an inputed image and extracts a feature map through ResNet, which is the backbone, and the extracted map outputs an objectness score and a bbox regressor through RPN. After that a 7 × 7 size feature map is outputted through the RoI Align process that rearranges the location of the region of interest, in order to deliver accurate location information of the objects. The 7 × 7 feature map predicts bbox regression and classification through fully connected layers, and predicts the mask through mask branches. The training objective of Mask RCNN is to reduce the loss of multitask as shown in equation (1).

\[ L_{\text{total}} = L_{\text{cls}} + L_{\text{bbox}} + L_{\text{mask}} \]

\[ L_{\text{cls}} : \text{Softmax Cross Entropy} \]
\[ L_{\text{bbox}} : \text{Regression} \]
\[ L_{\text{mask}} : \text{Binary Cross Entropy} \]

C. LIGHTWEIGHT DEEP LEARNING
The size and computation amount of deep learning models have been increasing for a better accuracy rate. However, as the model got bigger, problems emerged in the process of model training and inferencing, such as requirements for high computing resources and increased power consumption. There are several solutions that help increase GPU performance, such as distributed learning techniques or HW (Hardware) improvement methods. However, the HW method is inappropriate for making a lightweight model,
as small devices such as drones, robots, and smartphones can only mount limited types of computing. Therefore, various methods of applying a lightweight algorithm that can reduce the model size and computational costs have been proposed. There are four big methodologies in lightning deep learning, as shown in Table 1. First, pruning is a method that reduces parameters by removing neurons and their connections which do not have a significant influence on the model’s inference [30]–[33]. Quantization is a method that reduces the number of the bits of a parameter, by expressing the value of a learning parameter as a bit size with a low bit width [34]–[36]. Knowledge Distillation is a method of learning with one student model by using several pre-trained teacher models [37], [38]. Compact Network Design is a method to reduce the size of the model or the amount of computation by changing the structure of the neural network itself.

**TABLE 1.** Lightweight deep learning methods.

| Description                           | Pruning                          | Quantization                   | Knowledge Distillation | Compact Network Design |
|---------------------------------------|----------------------------------|--------------------------------|------------------------|------------------------|
| Removes weights or neurons below the reference value | Weight, activation, and gradient values are expressed in a low bit width | Lighten the pre-trained model | Lighten the network architecture, MobileNet [19], [27], ShuffleNet [39], DenseNet [40], EfficientNet [41] |

**D. MobileNet V2**

In this study, MobileNet V2 [19] was applied in order to reduce the weight of Mask RCNN. Spatial Convolution operation (hereafter Spatial Conv) in MobileNet V2 performs Depthwise Separable Conv (hereafter DW Conv). The way to operate Spatial Conv operation, as shown in Figure 2 (a) is to calculate the output of one pixel by performing $3 \times 3 \times 3$ kernel operation on the input image. However, in MobileNet V2, DW Conv was used as shown in Figure 2 (b). DW Conv is a conv that separates the input channels, performs a $3 \times 3$ kernel operation (Depthwise Conv) in each channel, concatenates the results and performs $3 \times 1$ kernel conv operation (Pointwise Conv) on it. This process reduces the amount of computation and the number of parameters to about 1/9 compared to the Spatial Conv operation, as shown in Table 2. In addition, MobileNet V2 reduced the amount of computation and parameters, by reducing the number of channels of convolution and using Inverted Residuals Block and Linear Bottleneck, which increases the number of channels only inside the block.

**TABLE 2.** Computation and no of parameters.

| Category | Spatial Conv | DW Conv |
|----------|--------------|---------|
| Computation | $H \times W \times 3 \times 3 \times N$ | $H \times W \times (3 \times 3 \times 3 \times 3 \times N)$ |
| Parameters | $3 \times 3 \times 3 \times N = 27N$ | $3 \times 3 \times 3 \times 3 \times N = 27 \times 3N$ |

**III. METHODOLOGY**

**A. RESEARCH PROCESS**

The procedure of the research is shown in 3. First, we collected pictures of warships among the publicly available Internet data. Then we made a ground truth dataset with the VIA (VGG Image Annotator) tool, so that the images can work as training data. After that, we checked the structure and the parameters of a base Mask RCNN and applied lightweight techniques to make a lightweight version. Finally, we evaluated the performance of the model after training it with the data and adding fine tuning.

**B. DATA COLLECT AND PROCESSING**

In order to produce the training data, we collected 2,156 pictures of naval warships of the United States, China, Japan, Russia, and S. Korea through online search. The collected data was divided into 1,819 training sets and 238 validation sets. Then, we used a VIA tool to draw the outlines on the warships that can also label them as battleships, aircraft carriers or auxiliary ships. Guns and guided missiles were labeled as well 4 (a). However, the number of labeled guns and missiles are not the exact amount of weapons on the warships, as only verifiable guns and missiles on the images were labeled. Each labeled data has the corresponding Instance Information, as it is shown in Figure 4 (b). We produced 5,078 ground truth datasets with this process.
C. LIGHTWEIGHT MASK RCNN

Our proposed framework is illustrated in Figure 5.

In order to make lightweight Mask RCNN, we modified Stage-1’s backbone and RPN. The default backbone of Mask RCNN is ResNet101 or 50, which are deep neural networks consisting of 101 or 50 layers as their names indicate. However, the backbone was replaced with MobileNet V2 to reduce the weight. In addition, the spatial convolution operation of RPN was replaced with DW Conv. The base RPN consists of 512 convolution kernels(with 3 × 3 filter). In order to maintain the performance of the RPN and lighten it, we composed the RPN of a spatial convolution operation with 256 kernels(with 3 × 3 filter) and a DW Conv with 256 kernels(with 3 × 3 filter). In addition, we added a pruning effect on an activation function by applying ReLU6. The RPN of the lightweight Mask RCNN is shown in Figure 6, and the parameters reduction are shown in Table 4.

D. PERFORMANCE MEASURE

We measured the performance of the model in two parts: accuracy and the degree of lightweight. First, accuracy was evaluated by two subdivided standards, the location prediction accuracy and the classification accuracy. The location prediction accuracy was evaluated based on IoU (Intersection of Union), which is the ratio of the predicted location area and the actual area of interest. Classification accuracy was evaluated by precision and recall, mAP (mean Accuracy Precision), which are calculated based on the Confusion Matrix. Precision is the ratio of those which were actually classified as True among those classified as True. And Recall refers to those which the model predicted as True among those that are actually True. mAP represents the area of the Precision-Recall graph. Each value can be obtained through the following equations (2), (3).

\[
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{2}
\]

\[
\text{mAP} = \frac{\sum_{i=1}^{N} \int_{0}^{1} P(r)dr}{N} \tag{3}
\]

TP is an outcome where the model correctly predicts the positive class. Similarly, a TN is an outcome where the model correctly predicts the negative class. A FP is an outcome where the model incorrectly predicts the positive class. And a FN is an outcome where the model incorrectly predicts the negative class.
The degree of lightweight, which is the second part of the model evaluation, was measured by comparing two models’ size and the total number of parameters. Also, the performance of the lightweight model was evaluated by the comparison of the 1-Epoch learning time with base model, which is the inference time of 1 sample image. The comparison of the accuracy rates in consideration of the number of parameters were used for the evaluation as well.

IV. RESULT AND ANALYSIS
We trained the base Mask RCNN under the same environment so that its operation can be compared with the one of lightweight model.

A. EXPERIMENT SETTING
The experiment environment and the major hyper parameters used on the model are shown in Table 5 and 6.

### TABLE 5. Experiment environment.

| Attribute Name | Value |
|----------------|-------|
| OS             | Window10 |
| CPU            | Intel i7-9750H@2.6GHz |
| GPU            | NVIDIA RTX-2060(6Gb) |
| Memory         | 32Gb |
| Tensorflow Ver | 1.4  |
| Keras Ver      | 2.08  |

### TABLE 6. Hyper parameter setting.

| Hyper Parameter       | Value |
|-----------------------|-------|
| Learning Rate         | 0.001 |
| Weight Decay          | 0.0001 |
| Min Detect Confidence | 0.7   |
| Epoch                 | 100   |
| Train Step            | 2,000 |
| Validation Step       | 200   |

After building the lightweight model, the image augmentation technique was applied to the data for the training. In this process, left and right inversion was arbitrarily made to simulate various angles within the images, and blurring was added to create the blurriness of images taken in various weather conditions such as fog, rain or snow. The data inputted in the training has passed through this process, so that the it would not be overlapped each time it was put in, in order to increase robustness and accuracy of the model.

B. MODEL PERFORMANCE

1) MODEL ACCURACY PERFORMANCE
In order to check the accuracy of the model, we selected 200 random pieces of data and evaluated the mean of IoU, Precision, Recall, and AP. As shown in Table 7 and Figure 7, meanIoU of the base Mask RCNN (which has ResNet101 as a backbone) was 0.88, meanPrecision of 0.94, meanRecall 1, and mAP 0.86. The proposed method showed meanIoU of 0.83, meanPrecision of 0.94, meanRecall 1, and mAP of 0.86. The proposed model’s classification accuracy is the same as that of the base Mask RCNN, but its mIoU was 5% lower compared to the base model. This means that the performance as a classification model is the same, but the localization performance of suggesting the region of interest is slightly lower. The proposed model modified the RPN to Depthwise Conv to lighten the weight. Because of that, the proposed model estimated the region of interest with about 0.74 million parameters while the base Mask RCNN estimated it with about 1.19 million parameters. This result can be analyzed that the difference in the number of parameters of the RPN leads to the difference in the region of interest.

### TABLE 7. Result of experiment about accuracy.

| Category      | Base Mask RCNN | The Propose Lightweight Mask RCNN |
|---------------|----------------|-----------------------------------|
| mIoU          | 0.88           | 0.83                              |
| mPrecision    | 0.94           | 0.94                              |
| mRecall       | 1.00           | 1.00                              |
| mAP           | 0.86           | 0.86                              |

In order to check what happens because of 5% of localization error, we compared the location accuracy of two model’s outputs that came out from the same input image. The output result is as shown in Figure 10, and there are slight masking differences in the bow and the stern. To summarize this results in terms of the accuracy performance, the proposed model has about 5% of performance degradation in estimating the region of interest, while the classification accuracy is the same compared to the base model.

2) MODEL LIGHTWEIGHT PERFORMANCE
Figure 9 and Table 8 show the model size, total number of parameters, 1 Epoch training time, 1 sample image Inference time and mAPs per total parameters of the base model and the proposed model. As shown in Table 8, the number of parameters of the proposed model and the size were reduced by 65% compared with the base model. As it became lighter, the training time for one epoch was reduced by 11.6% and the image inferenceing time by 46%. This means that the proposed model is able to learn and infer quickly.
TABLE 8. Result of experiment about lightweight.

| Category              | Base Mask RCNN | The Proposed Lightweight Mask RCNN |
|-----------------------|----------------|-----------------------------------|
| Model Size            | 244Mb          | 85.4Mb(65%)                      |
| No.of Parameters      | 63,754,934     | 22,283,686(65%)                  |
| Time Epoch Training   | 4,448s         | 3,934s(11.6%)                    |
| Inference Time        | 0.8268s        | 0.562s(46%)                      |
| mAP/Parameters        | 1.4351B-8      | 3.8455B-8(167%)                  |

TABLE 9. FLOPS comparison of the two models.

| Input Size       | Base Mask RCNN | The Proposed Lightweight Mask RCNN |
|------------------|----------------|-----------------------------------|
| 256×256          | 27.22          | 7.269                             |
| 512×512          | 108.86         | 29.07                             |
| 768×768          | 244.926        | 65.42                             |
| 1024×1024        | 435.4          | 116.301                           |

FIGURE 9. Lightweight performance comparison.

C. ANALYSIS OF MODEL OPERABILITY BY COMPUTING ENVIRONMENTS

In order to examine whether the proposed model can be applied in the actual computing environment, we checked FLOPS (FLoating point Operations Per Second) for each input size of two models respectively. FLOPS is the number of floating-point operations that a computer can perform in 1 second. Therefore, larger FLOPS means the larger amount of calculation capability. Peak FLOPS of commercial CPUs or GPUs is the maximum computation amount of the processing unit, and the higher the number is, the faster the amount and speed of computation are. Table 9 shows the respective FLOPS of base model and the proposed model by each input size. As resolution of the input image got higher and the image size got bigger, FLOPS increased proportionally. This means that a high performance computing resource is required for the model to handle images with high resolutions.

We checked the performance specifications of commercial CPUs and GPUs that meet the FLOPS conditions. First, in terms of CPU, the Intel CPU i9-7980XE@ 2.60GHz which was released in 2017, shows a GFLOPS of 140.43. In terms of GPU, the NVIDIA GeForce GT 710 model released in 2013, shows the performance of 366 in its GFLOPS. Therefore, the proposed model can operate in a computing environment with specifications of i9-7980XE and GeForce GT 710 or higher.

TABLE 10. Statistics of the final test result.

| Image Condition | Top View | Bow View | Asten View | Port View | Stbd View | Dark | Cloudy | Multiple Ships |
|-----------------|----------|----------|------------|-----------|-----------|------|--------|----------------|
| Target Class    | Aircraft Carrier | Battleship, Gun | Battleship | Battleship, Gun, Missile | Battleship, Gun, Missile | Battleship, Gun, Missile | Aircraft Carrier Battleship, Gun |
| Target Quantity | 1        | 2        | 1          | 2         | 2         | 4    | 3      | 5              |
| Detected Target Quantity | 1       | 2        | 1          | 2         | 2         | 4    | 2      | 3              |
| Missing Target  | X        | X        | X          | X         | X         | X    | Missile 1 | Gun 2          |
| Detection Accuracy | 0.94    | 0.9, 0.96 | 0.995      | 0.995, 0.99 | 0.962, 0.998 | 0.976, 0.998, 0.961, 0.904 | 0.924, 0.996 | 0.968, 0.904, 0.824 |
| View               | Original Image                                                                 | Predicted Image                                                                 |
|--------------------|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Top View           | ![Top View Original](image1) ![Top View Predicted](image2)                    | ![Top View Predicted](image3) ![Top View Predicted](image4)                    |
| Bow and Astern View| ![Bow and Astern View Original](image5) ![Bow and Astern View Predicted](image6) | ![Bow and Astern View Predicted](image7) ![Bow and Astern View Predicted](image8) |
| Stirb View         | ![Stirb View Original](image9) ![Stirb View Predicted](image10)               | ![Stirb View Predicted](image11) ![Stirb View Predicted](image12)               |
| Port View          | ![Port View Original](image13) ![Port View Predicted](image14)                | ![Port View Predicted](image15) ![Port View Predicted](image16)                |
| Dark View          | ![Dark View Original](image17) ![Dark View Predicted](image18)                | ![Dark View Predicted](image19) ![Dark View Predicted](image20)                |
| Cloudy View        | ![Cloudy View Original](image21) ![Cloudy View Predicted](image22)            | ![Cloudy View Predicted](image23) ![Cloudy View Predicted](image24)            |
| Multiple Ships     | ![Multiple Ships Original](image25) ![Multiple Ships Predicted](image26)      | ![Multiple Ships Predicted](image27) ![Multiple Ships Predicted](image28)      |

**TABLE 11.** The final test results ((a): original images, (b): predicted images).
However, this specification is deficient in running base Mask RCNN, and problems such as lack of storage space and slow process speed might occur if it is run under this circumstance.

D. FINAL TEST RESULT

To verify the performance of the proposed model, we operated the final test with new images that were not used in the processes of training and validation. In the test we examined the operation of the model under these conditions: viewpoint of looking at the warship (top view, front/rear view, left/right view), different weather (dark/cloudy), presence of multiple/multi-class warships. As shown in Table 10, the model failed to detect guided missiles and guns in the condition of Cloudy and Multiple Ships. It detected and classified warships, guided missiles, and guns accurately in case of having high resolution input images such as Top View and Stbd View. However, the identification rate got lower in case the shape of the ships were not clearly identifiable because of low resolution caused by the Cloudy or Multiple Ships conditions, or because the warships were shot from a far distance. Still, the model showed 0.9 or higher degree of classification accuracy among the identified images, proving its capability in the performance of classification. The additional studies are needed in order to solve the problem of a low identification rate for distant or low-resolution images, such as adding an image preprocessing network or converting resolution.

E. APPLYING CONCEPT TO NAVAL WARSHIP

We want to suggest how to apply the proposed model to the naval warship. In order to operate the model, following components are required: remote sensor, image processing unit, and an indication unit that can display the result of the process. The naval warship is equipped with a high-performance electro-optical targeting system (EOTS) that can detect remote objects. In addition, the image shot by the optical camera (EO/IR, Electro Optical / InfraRed) mounted on a drone or an UAV can be transmitted through the wireless network in real time to the warship’s computer room. The transmitted image is used as an input of the proposed model. After that, Laptop with GPU (or Desktop with GPU) processes it in real time and displays the result. The proposed model is capable of providing quick and accurate information about the threat priority, and it will help a commander to make a judgment over the situation and make a command decision faster. Figure 11 shows the conceptual diagram of the naval warship’s operation that the proposed model was applied on.

V. CONCLUSION

In this study, we proposed a lightweight deep learning model that can identify various high value warships in video. By operating the proposed model, we could classify warships as battleships, auxiliary ships, and aircraft carriers, and identify each warship’s armament. The mAP, which is the classification performance of a model, was 0.86, and the mIoU, the performance of determining the location, was 0.83. The weight of the model was reduced to about 64% than base Mask RCNN, which leads it to spend shorter time in learning and inference.

The contributions of this study are as follows. First, we made the lighter version of the Mask RCNN model so that it can be operated in equipment with low-performance computing resources. Second, we developed a model that can automatically classify classes of unidentified warships during naval operations and thereby help a commander. Third, we proposed computing resources that are required for the application of the proposed model and the method to apply the model on naval warships.

However, further studies are needed as the model needs improvement of recognition in low-resolution images. First, a bigger dataset should be made to train the model to regularize the features of various warship images. Second, input images pre-process networks should be developed in order to modify low resolution image to have higher resolution. Finally, the model will become more useful if it is trained using additional data to identify different platforms, such as aircraft and drones etc. We will address these in our future work.

REFERENCES

[1] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” Handbook Brain Theory Neural Netw., vol. 3361, no. 10, pp. 1–14, 1995.

[2] C. P. Langlotz, B. Allen, B. J. Erickson, J. Kalpathy-Cramer, K. Bigelow, T. S. Cook, A. E. Flanders, M. P. Lungren, D. S. Mendelson, J. D. Rudie, G. Wang, and K. Kandarpa, “A roadmap for foundational research on artificial intelligence in medical imaging: From the 2018 NIH/RSSNA/ACR/the academy workshop,” Radiology, vol. 291, no. 3, pp. 781–791, Jun. 2019.
