Evaluating Lightweight Object Detection Models for Lesion Detection

Talal Siddiqui **Zeyu Wang**
Northeastern University
Boston, MA
siddiqui.t@northeastern.edu wang.zeyu8@northeastern.edu

Abstract

In the past decade, object detection models have made massive gains in performance [1]. Such systems have also performed well on medical datasets [1], opening a door for their use in hospitals and clinics, as a tool for medical providers to easily catch abnormalities in medical imaging. But, the training and inference process of many of these models require a considerable amount of computational and economic resources [2], which may not be available to many in developing countries. So, we explore how medical providers in developing countries may use lightweight object detection models for detection lesions in CT scan images. We do this by evaluating the performance of two configurations of pre-trained YOLOv5 object detection models on a lesion dataset called DeepLesion, with one of the configurations being modified with SEBlocks and Mish activation functions. We compare these models with heavier, state-of-the-art models for this dataset and do a careful examination of why one may perform better than the other. Our code for the paper is located at https://github.com/talals1/cs5330-final-project/

1 Introduction

The past decade or so has seen an explosion of better and more powerful deep learning models emerge [1]. Among them is a class of models called *object detectors* [1]. The job of these models is to detect objects in images and draw a bounding box as to where the model thinks that object is in the image. The state-of-the-art (SOTA) object detection models geared towards medical datasets often have large and complex architectures and require a large computational power for training and inference, especially in terms of resources like GPUs, RAM, power, and more [7].

But, there are many places and countries around the world where medical providers do not have the economic and computational resources to train and utilize such models [2]. Today, 6 billion people live in developing countries [2]. In 2019, it was estimated that 780 million people do not have access to reliable electricity [4]. It can be understood that the medical care received by these people could be enhanced by the assistance of machine learning models, in particular those that can detect objects in medical imaging [4]. For example, doctors could utilize a tumor detection model for flagging potential tumors in CT scans, potentially increasing the early detection rate and in turn survival rate.

Thus, the goal of this project was to evaluate how well a simple object detection model, in our case YOLOv5, can perform on a lesion detection dataset.

Computer vision (CV) is a task that requires complex models because the world we live in is inherently complex [3]. It is full of patterns, colors, and meanings that are only understood by many layers of cultural, societal, and historical knowledge [3]. In order to capture and predict patterns from the rich information encapsulated in snapshots of our world, our systems and models must also be able to represent those complex relationships, which is why CV problems use architectures like residual networks (ResNets) [6], convolutional neural networks (CNNs) [6], and more recently Vision Transformers (ViT) [11].

To evaluate how well a lightweight object detection model performs in restricted conditions when using it in a medical setting, we will train such a model (YOLOv5) with a lesion detection dataset called *DeepLesion*. More specifically, we will train a lighter version of YOLOv5 called YOLOv5n and compare it to a heavier version called YOLOv5x that also has other modifications. From there, we will evaluate its performance and compare it to SOTA in terms of both performance and efficiency.

Our primary goal is to evaluate the trade-offs between model performance and computational efficiency. Specifically, we aim to answer the following questions:

- How well does the lightweight YOLOv5n perform on lesion detection tasks in low-resource settings?
- How does the high-capacity YOLOv5x compare in terms of accuracy and generalization?
- What improvements can be made by incorporating techniques like SEBlock and Mish activation into these models?
- How do these models compare against SOTA methods such as MULAN and AlignShift on the DeepLesion dataset?

By addressing these questions, this study contributes to the ongoing efforts to make advanced medical imaging tools more accessible and efficient, particularly in regions where resources are limited.

Acknowledgment of Open-Source Code
This study utilized the open-source YOLOv5 implementation provided by Ultralytics (https://github.com/ultralytics/yolov5) as the foundation for our experiments. PyTorch (https://pytorch.org) was used as the primary deep learning framework for training and evaluation. Both tools significantly accelerated development and experimentation, and we acknowledge their contributions to this work.

2 Related Work

M.G. Ragab et al. have done a comprehensive survey of the usage of YOLO for medical object detection from 2018 to 2023 [5]. It notes that YOLO has seen wide adoption due to its real-time performance, efficiency, and accuracy helped medical providers make timely and accurate decisions. The paper identifies more than 30 publications using YOLO for medical object detection, showing that there is indeed an interest in using YOLO for this purpose. But, the paper admits that though YOLO has its strengths, it comes at a cost, as the model requires large datasets for training and has problems with small objects and occluding/intersecting bounding boxes.

In a more general overview, De-Arteaga et al. explores how machine learning in developing countries presents new challenges and opens doors to possibly novel solutions [2]. The paper describes how the applying ML models in developing countries can sometimes be difficult as the "availability of data, computational capacity, and Internet accessibility are often markedly more limited than in developed countries." As well, it attempts to formalize a concrete definition of what Machine Learning for Developing Countries (ML4D) is and goes into a literature review of the most prominent literature across topics like health care, institutional, economics, sociology, and environmental concerns. The most relevant to us is the ML health care research in developing countries, for which the papers highlight research such as automated diagnosis systems for urban hospitals.

Mittal's work is a comprehensive survey of lightweight object detection models that are meant for edge devices [6]. It defines the "edge" in edge computing as "a component of an architecture of distributed computing where data processing resides near the edge where devices or individuals generate or consume that data." It emphasizes how running deep learning models on devices on the edge has benefits like enhancing data privacy by making sure the data never has to be uploaded somewhere else for inference, saving bandwidth due to the aforementioned lack of uploading any data, and most importantly, can potentially offer much faster results by running the model locally. It highlights models such as YOLO, MobileNet, ShuffleNet, and more as suitable lightweight detection models. For our work, medical providers are definitely at the "edge" because giving them the ability to locally run lightweight object detection models on inexpensive hardware (say, an Android phone) has the potential to tremendously increase the rate of early detection of abnormalities in medical scans/imaging.

During the height of the COVID-19 epidemic in 2022, a paper by Ukwandu et al. evaluated the usefulness of lightweight object detection models for precisely detecting COVID-19 infection via medical imaging, particularly chest CT scans [7]. They utilized the MobileNetV2 model pre-trained on the ILSVRC-12 challenge ImageNet dataset with around 3.5 million parameters and found that it delivered comparable performance to heavier models such as VGG16 and VGG19, each of which had more than 100 million parameters. As well, they highlighted how MobileNet, a neural network designed specifically for efficiency, would be adequate for clinicians in low and middle-income countries who may not have the resources for running larger models to aid them in diagnosis.

One of the examples of YOLO being applied to medical datasets that was mentioned in Mittal's work was the work by Pacal et al. in 2021 who used YOLOv4 for colonic polyp detection [8]. They describe colonic polyps as tumors that appear in the colon, which then lead to colorectal cancer. Their proposed method of modifying YOLOv3 to utilize CSPNet, SiLU activation function, and advanced data augmentation techniques helped them surpass stock YOLOv3 and YOLOv4 on the SUN and PICCOLO polyp detection datasets.

Lightweight neural networks have been explored for other uses than just object detection. In a paper by He et al from 2018, the team explored the utility of light weight machine learning models for detecting arrhythmias [9]. They propose a model called LiteNet, a lightweight neural network that outperforms comparable models in diagnosing arrhythmias using echocardiogram (ECG) data, while being suitable for deployment on mobile devices due to its low memory and computational requirements. This shows that lightweight machine learning models in general have a lot of potential use cases, with one of the target platforms being mobile devices due to their ubiquity in all countries.

3 Method

For our approach, we utilized an open source implementation of the YOLOv5 object detection model which is developed by an organization called Ultralytics. Ultralytics offers multiple versions/configurations of YOLOv5, from least to most powerful. The API was very simple to use and allowed us to easily finetune the models on our data.

The first configuration we chose was YOLOv5n, short for YOLOv5 Nano and is the smallest configuration. According to Ultralytics, this configuration has the fastest inference time of 73.6 ms on a CPU, but also has the smallest number of parameters (2.6 million) and has the worst mAP@50 score of 0.343 on the validation dataset [10].

The second configuration we chose was YOLOv5x, short for YOLOv5 Extra Large. This is one of the more larger configurations and boasts around 97 million parameters and has a 763.2 ms inference time on a CPU [10]. It has a mAP@50 score of 0.532 on the validation dataset. [10] We then modified this model to squeeze and extract blocks (SEBlocks) and replaced the default ReLU activation function with the Mish activation function. SEBlock (Squeeze-and-Excitation Block) was integrated to recalibrate channel-wise features, helping the model focus on the most informative regions in the images [12]. Additionally, the Mish activation function, a smooth and non-monotonic alternative to ReLU, was incorporated to improve gradient flow and enhance feature richness [13]. The Mish activation function was developed by Diganta Misra [13] and is defined as

$$f(x) = x \tanh(softplus(x))$$

where $softplus(x) = \ln(1 + e^{x})$ [13].

In order to save time on training, we utilize pre-trained weights for YOLOv5n and then finetuned the model on the DeepLesion dataset. This is an example of transfer learning, which allows us to use models trained on general data to then use its knowledge of patterns to detect structure in a different but similar context [14].

4 Experiments

4.1 Dataset

To test our models, we used the DeepLesion dataset, which is a collection of 32,735 lesions in 32,120 CT slices from 10,594 studies of 4,427 unique patients [15]. It was compiled by researchers at the National Institutes of Health (NIH) who mined their medical center's picture archiving and communication systems (PACS) for CT scans and their annotations [15]. There are 10 types of lesions in the dataset: bone, abdomen, mediastinum, liver, lung, kidney, soft tissue, pelvis, and other, where "other" could be anywhere else in the body [15].

4.2 Preprocessing

The bounding box data in the DeepLesion dataset is in $x1, y1, x2, y2$ format, when $(x1, y1)$ is the top left corner of the bounding box and $(x2, y2)$ is the bottom right corner [15]. For the annotation data to be

understood by the Ultralytics YOLO model, we had to convert all bounding box annotations in DeepLesion into x-center, y-center, the width and height, with x-center and width being normalized using the image width (512 pixels) and y-center and height being normalized using the image height (also 512 pixels) [15].

4.2 Setup and Experiments

Due to storage restrictions, we were unable to use the full dataset, which is around 220 GB [15]. Instead, our experiments used subsets around the size of 7000-8000 scans.

For comparison, we consider the "state-of-the-art" to be the MULAN model (multitask universal lesion analysis network) developed by Yan et al in 2019 [16] and the AlignShift model coupled with bounding map conditioning developed by Li et al in 2021 [17]. One notable characteristic that makes these models different from our approach is that they use the 3D geometric data of the lesion provided in the DeepLesion dataset for object detection [15, 16, 17], rather than simple training on the raw CT scan and annotation data like our method does.

We will be evaluating our models with 3 metrics: precision, recall/sensitivity, and mAP@50. Precision measures how many results returned were actually true/relevant, while recall measures how many true positives were actually returned as positive. The mean average precision calculated at an intersection over union (IoU) threshold of 0.50, AKA mAP50, measures how well a computer vision model predicts the bounding boxes for the objects in a given image.

We had 4 experiments:

1.) How well does the smallest YOLOv5 model (YOLOv5n) perform with minimal changes?
2.) How well does the largest YOLOv5 model (YOLOv5x) perform as a baseline for general lesion detection?
3.) How well does the YOLOv5x model perform with the integration of SEBlock?
4.) How well does the YOLOv5x model perform with Mish activation replacing the default SiLU?

4.2.1 Experiment 1: Lightweight Detection with YOLOv5n

For our first experiment, we trained YOLOv5n on Google Colab for 70 epochs with a batch size of 150 on a NVIDIA T4 GPU with 16GB VRAM. The subset of data used had 6988 images, with 5590 for training and 1398 for validation/testing. It had all lesion types represented but the "other" type was overrepresented as it was 75% of images in this subset.

4.4.2 Experiment 2: Baseline Detection with YOLOv5x

The second experiment aimed to establish a baseline for general lesion detection using YOLOv5x, the largest configuration of YOLOv5. A subset of the DeepLesion dataset was created with 8,078 training images, 1,277 validation images, and 1,696 test images. Unlike Experiment 1, this subset focused solely on the "other" lesion type to simplify the detection task.

Training was performed locally on an NVIDIA RTX 4090 GPU with 64GB of RAM. The following command was used to train the model:

```
python train.py --data deeplesion.yaml --weights yolov5x.pt --epochs 50 --img 640 --batch-size 16
```

4.4.3 Experiment 3: Integrating SEBlock

In the third experiment, SEBlock (Squeeze-and-Excitation Block) was integrated into the YOLOv5x architecture to enhance channel-wise feature recalibration. Everything else was the same Experiment 2.

4.4.4 Experiment 4: SEBlock + Adding Mish Activation

The final experiment introduced the Mish activation function into the YOLOv5x model, replacing the default SiLU activation. Mish's smooth, non-monotonic properties were expected to enhance gradient flow and improve the model's ability to learn richer feature representations. Everything else was the same as Experiment 2.

4.3 Results

Much of our results showed our models underfit the dataset by a substantial amount when compared to the SOTA.

Our smallest model, the YOLOv5n, had an average Recall/Sensitivity of 0.22, a Precision of 0.12, and a mAP50 score of 0.10.

Our largest model, YOLOv5x, was able to get a considerable boost in Precision when using SEBlock and the Mish activation function, bumping it from 0.45 to 0.57. But that boost came at a cost because it decreased the Recall from 0.34 to 0.29 and the mAP50 score from 0.32 to 0.29.

We assume the reason that our models ran poorly due to the limited number of epochs the models we trained on. Due to computation constraints, the YOLOv5n and YOLOv5x variations were trained at low numbers of epochs. It was especially difficult to train YOLOv5n as it was trained on Google Colab, and the team encountered several problems with Google flagging us as a robot and banning us for certain periods of time. As well, larger epochs can substantially increase the training time, so we tried to mitigate that by increasing the batch sizes.

We also assume that the relatively small subset of data may have played a factor in the poor performance, as YOLO has been noted to be needing a substantial amount of data in order to have good performance metrics.

YOLOv5x had higher scores than YOLOv5n, which we deduce is from the single lesion subset of DeepLesion coupled with 50 times more parameters, giving it the ability to represent complex relationships more easily.

YOLOv5n's class-specific metrics seem interesting, as they show excellent performance on lesion types such as "other" with precision of 0.308, recall of 0.331, and mAP50 of 0.202. Though YOLOv5x still outperforms YOLOv5n in this regard, it is interesting to observe that they are almost comparable. The YOLOv5n model also did better than its average on the "mediastinum" and "lung" lesion types. The classes YOLOv5n did the best on also had the most representation in the subset of data we chose to train it on, lending more evidence to the theory that YOLO models may work better when given a large amount of data.

MULAN and AlignShift with modifications by Li et al are still the leaders of this dataset in this niche of object detection, having Recall scores of 0.8522 and 0.8716 respectively. Other metrics like Precision and mAP@50 were not provided for those models. As mentioned before, their architecture allows them to utilize the 3D geometry data of lesions available in the DeepLesion dataset [15], but our

YOLO models simply take the raw CT scan data along with the bounding box information to train and perform inference.

Model	Precision	Recall	mAP50
MULAN	?	0.8522	?
AlignShift (Li et al)	?	0.8716	?
YOLOv5n	0.12	0.22	0.10
YOLOv5x baseline	0.45	0.34	0.32
YOLOv5x + SEBlock	0.46	0.33	0.315
YOLOv5x + SEBlock + Mish	0.57	0.29	0.29

Table 1: Comparing our models average precision, recall/sensitivity, and mAP50 scores with the SOTA

Class	Images	Instances	P	R	mAP50	mAP50-95
all	1397	1434	0.143	0.141	0.108	0.0608
other	1039	1068	0.308	0.331	0.202	0.11
bone	2	2	0	0	0.049	0.0163
abdomen	72	73	0.0879	0.0274	0.0497	0.0279
mediastinum	100	101	0.341	0.327	0.242	0.128
liver	55	56	0	0	0.0021	0.000374
lung	56	57	0.261	0.316	0.218	0.118
kidney	10	10	0	0	0.00799	0.00425
soft_tissue	22	22	0.0952	0.0909	0.045	0.0304
pelvis	43	45	0.194	0.178	0.159	0.113

Table 2: YOLOv5n results per class

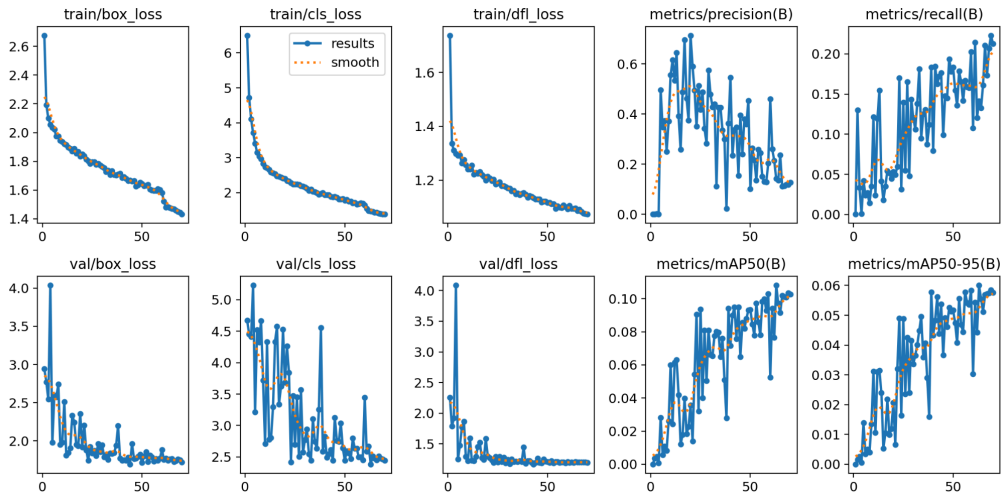

Figure 1: Performance metrics of YOLOv5n

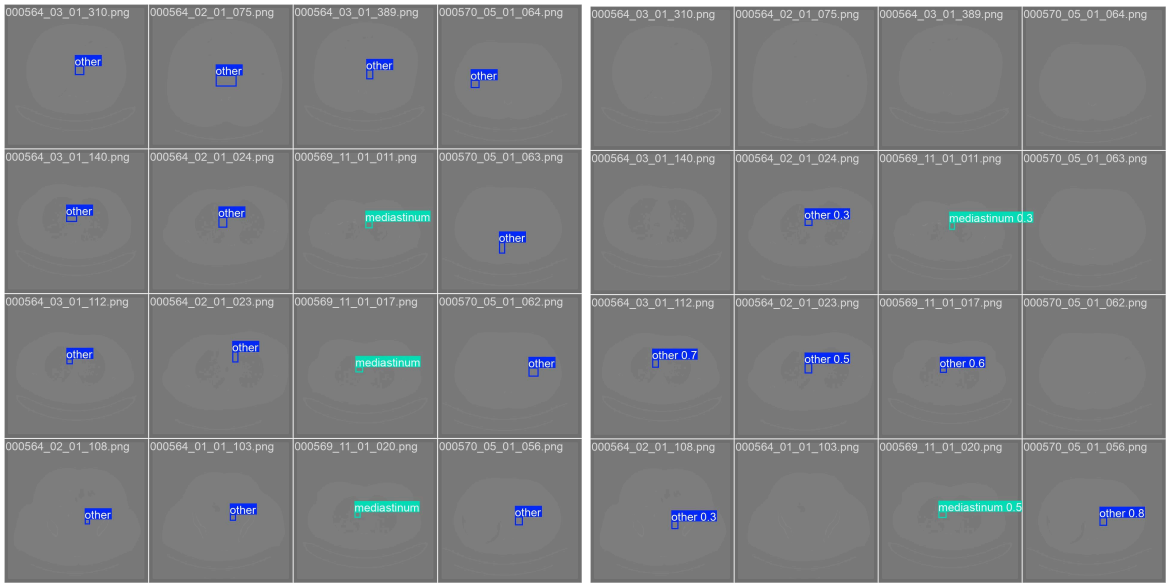

Figure 2: Here is a sample prediction from our YOLOv5n model in training. The left image is a mosaic of true annotations for a set of 16 images. The right image is a mosaic of annotations predicted by our model. As you can see, it did not predict anything for 8 images and had the wrong label for one of the images.

5 Conclusion

As it can be seen, there is a lot of potential for lightweight object detection models like YOLO to be used for object detection in medical settings, especially in those areas where computational and economic resources are limited. But as we saw in our experiments, it can be challenging to finetune both small and large YOLOv5 configurations on medical images, and it is possible that our test environments and

parameters were not optimal. Due to a lack of time and computational resources, we were unable to explore this further, but we hope to come back one day to improve our performance.

5.1 Further Work

Though we were fine tuning pretrained models that are supposed to be lightweight, they still required substantial computational power. This is simply not optimal for those who do not have such resources or are unable to use cloud-based versions of ML training platforms. Because of this, we encourage more research to be done on how to make machine learning models that can fully take advantage of inexpensive hardware. One example is Android phones, which are ubiquitous throughout the world. Further research of using Android phones as a testbench for evaluating object detection performance is something we really recommend as it shows a real world use case that is practical and easy to simulate.

Trying out object detection models other than YOLO is also a strong recommendation from the team. Models such as MobileNetV3 [20], EfficientDet [21], NanoDet [22], and more have been designed in a similar vein as YOLO, but in some cases are even more efficient at the cost of accuracy [5]. Testing a diverse range of lightweight object detection models would be great in showing medical providers in developing countries as to which models would be best suited for them given the amount of resources available to them.

Locally training and performing inference may not be an option to all, so an alternative that needs more exploration is cloud-based runtime environments. A plethora of them have flooded the market over the past few years, with Google Colab and Kaggle being the most well known, with newcomers such as Vast.ai, Lambda Labs, etc.. Many of these platforms offer free but restricted GPU usage, which truly lowers the barrier to those unable to afford their own system. But, we encourage more research to be done to compare and contrast different platforms in terms of compute offered, pricing, storage, and more. Cloud-based environments may even be optimal for mobile clinics.

Overall, lightweight deep learning models have a tremendous amount of potential which is sadly untapped. Encouraging developers to create such lightweight systems could possibly help medical providers in low and middle income countries give better care for their patients, whether it be earlier detection of tumors or lesions or helping detect arrhythmia. In conclusion, we hope the next generation of researchers and developers put more thought into lightweight models so that all of humanity can benefit from the wonders of machine learning.

References

[1] Zaidi, Syed Sahil Abbas, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. "A survey of modern deep learning based object detection models." Digital Signal Processing 126 (2022): 103514. https://arxiv.org/abs/2104.11892

[2] Maria De-Arteaga, William Herlands, Daniel B. Neill, and Artur Dubrawski. 2018. “Machine Learning for the Developing World.” ACM Trans. Manage. Inf. Syst. 9, 2, Article 9 (June 2018), 14 pages. https://doi.org/10.1145/3210548

[3] McDonough, Molly. “The Limits of Computer Vision, and of Our Own.” Harvard Medicine Magazine, May 1, 2024. https://magazine.hms.harvard.edu/articles/limits-computer-vision-and-our-own.

[4] Hannah Ritchie, Pablo Rosado and Max Roser (2019) - “Access to Energy” Published online at OurWorldinData.org. Retrieved from: 'https://ourworldindata.org/energy-access' [Online Resource]
[5] Ragab, Mohammed Gamal, Said Jadid Abdulkader, Amgad Muneer, Alawi Alqushaibi, Ebrahim Hamid Sumiea, Rizwan Qureshi, Safwan Mahmood Al-Selwi, and Hitham Alhussian. "A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023)." IEEE Access (2024). https://ieeexplore.ieee.org/document/10494845
[6] Mittal, P. “A comprehensive survey of deep learning-based lightweight object detection models for edge devices.” Artif Intell Rev 57, 242 (2024). https://doi.org/10.1007/s10462-024-10877-1
[7] Ukwandu, Ogechukwu, Hanan Hindy, and Elochukwu Ukwandu. "An evaluation of lightweight deep learning techniques in medical imaging for high precision COVID-19 diagnostics." Healthcare Analytics 2 (2022): 100096. https://doi.org/10.1016/j.health.2022.100096
[8] Pacal, Ishak, Ahmet Karaman, Dervis Karaboga, Bahriye Akay, Alper Basturk, Ufuk Nalbantoglu, and Seymanur Coskun. "An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets." Computers in biology and medicine 141 (2022): 105031. https://doi.org/10.1016/j.compbiomed.2021.105031
[9] He, Ziyang, Xiaoqing Zhang, Yangjie Cao, Zhi Liu, Bo Zhang, and Xiaoyan Wang. 2018. "LiteNet: Lightweight Neural Network for Detecting Arrhythmias at Resource-Constrained Mobile Devices" Sensors 18, no. 4: 1229. https://doi.org/10.3390/s18041229
[10] Ultralytics. “Ultralytics YOLOv5.” YOLOv5 - Ultralytics YOLO Docs, November 7, 2024. https://docs.ultralytics.com/models/yolov5/.
[11] Shehzadi, Tahira, Khurram Azeem Hashmi, Didier Stricker, and Muhammad Zeshan Afzal. "Object Detection with Transformers: A Review." arXiv (2023). https://doi.org/10.48550/arXiv.2306.04670
[12] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141. 2018. https://arxiv.org/abs/1709.01507v4
[13] Misra, Diganta. "Mish: A self regularized non-monotonic activation function." arXiv preprint arXiv:1908.08681 (2019). https://arxiv.org/abs/1908.08681
[14] Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. "A comprehensive survey on transfer learning." Proceedings of the IEEE 109, no. 1 (2020): 43-76. https://arxiv.org/abs/1911.02685
[15] Yan, Ke, Xiaosong Wang, Le Lu, and Ronald M. Summers. "DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning." Journal of medical imaging 5, no. 3 (2018): 036501-036501. https://pubmed.ncbi.nlm.nih.gov/30035154/
[16] Yan, Ke, Youbao Tang, Yifan Peng, Veit Sandfort, Mohammadhadi Bagheri, Zhiyong Lu, and Ronald M. Summers. "MULAN: multitask universal lesion analysis network for joint lesion detection, tagging, and segmentation." In Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part VI 22, pp. 194-202. Springer International Publishing, 2019. https://arxiv.org/abs/1908.04373v1
[17] Li, Han, Long Chen, Hu Han, Ying Chi, and S. Kevin Zhou. "Conditional training with bounding map for universal lesion detection." In International Conference on Medical Image

Computing and Computer-Assisted Intervention, pp. 141-152. Cham: Springer International Publishing, 2021. https://arxiv.org/abs/2103.12277v1

[18] Yang, Jiancheng, Yi He, Xiaoyang Huang, Jingwei Xu, Xiaodan Ye, Guangyu Tao, and Bingbing Ni. "AlignShift: bridging the gap of imaging thickness in 3D anisotropic volumes." In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23, pp. 562-572. Springer International Publishing, 2020. https://arxiv.org/abs/2005.01969

[19] "DeepLesion Benchmark Leaderboard." https://paperswithcode.com/sota/medical-object-detection-on-deeplesion

[20] Howard, Andrew, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang et al. "Searching for MobileNetV3." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 1314-1324. 2019. https://doi.org/10.48550/arXiv.1905.02244

[21] Tan, Mingxing, Ruoming Pang, and Quoc V. Le. "EfficientDet: Scalable and Efficient Object Detection" In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10781-10790. 2020. https://doi.org/10.48550/arXiv.1911.09070

[22] RangiLyu. "NanoDet-Plus." GitHub. Accessed December 10, 2024. https://github.com/RangiLyu/nanodet.