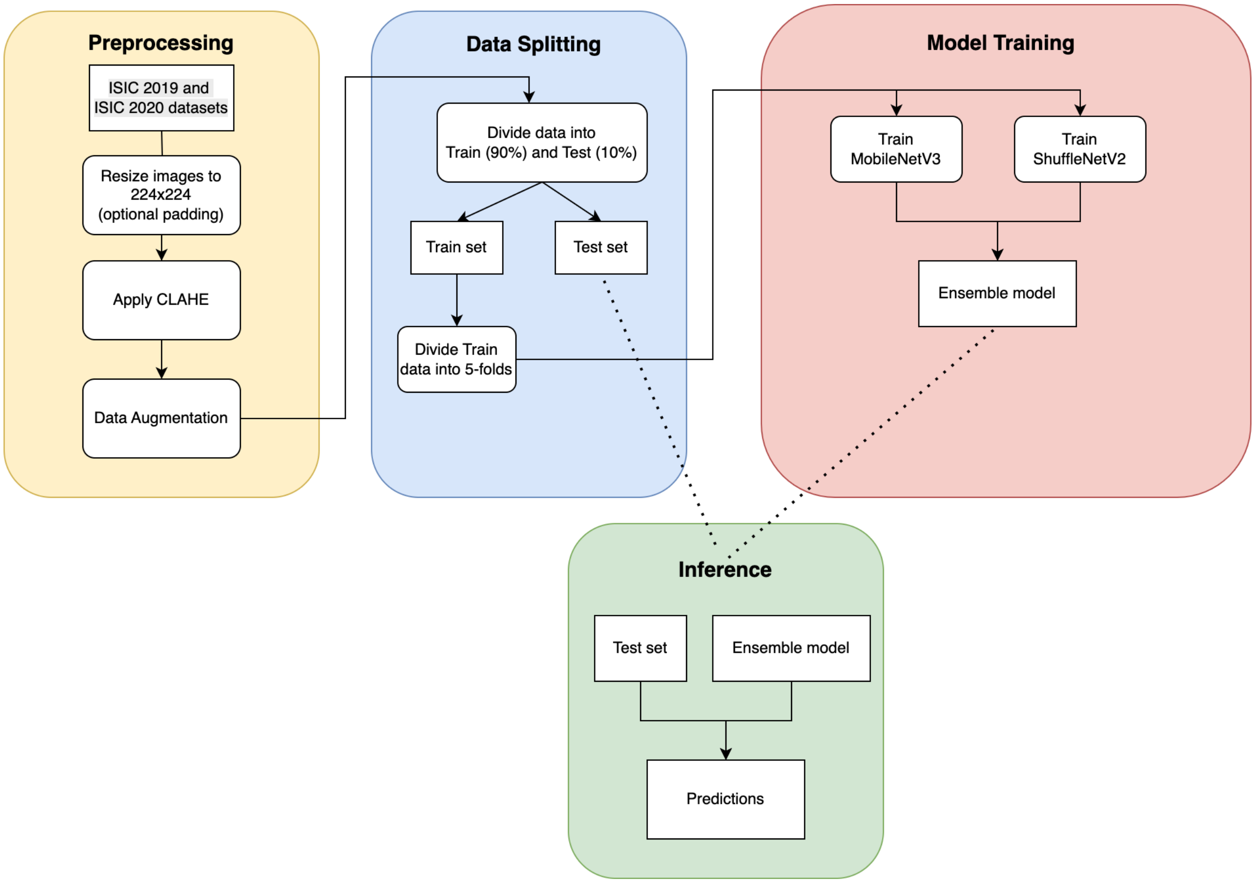
Methodology documentation

1. Overview

The goal of this project is to classify skin lesion images into two categories — benign and malignant — using an ensemble of convolutional neural networks. The overall project architecture is illustrated in the figure below. This document provides an overview of the methodologies employed throughout the project. For a detailed report on model performance, dataset structure, and additional information, please refer to the notebooks on this link.



2. Dataset overview

The dataset is composed of high-quality dermatological images, labeled based on histopathological diagnosis. The ISIC archive has been widely used in previous research, including in dermatologist-level skin cancer classification tasks (Esteva et al., 2017), making it a standard benchmark for evaluating melanoma detection models. The dataset consists of a combination of ISIC 2020 and ISIC 2019 datasets. The list of images can be found when getting the data inside the “merged_labels.csv”. The exploratory data analysis can be found in the notebook *initial_eda.ipynb* on the drive link.

3. Image preprocessing pipeline

3.1. Padding and resizing

Padding and resizing the images to a uniform size of 224 by 224 pixels is essential for feeding them into standard convolutional neural network architectures, many of which expect inputs of that specific resolution. Resizing alone can distort the aspect ratio and alter lesion shape—potentially misleading the model. By padding images first to make them square (while preserving the original aspect ratio), and then resizing, we avoid such distortions. This ensures that lesions retain their geometric properties, which are often diagnostic (e.g., irregular borders or asymmetry). This resizing approach is consistent with best practices in medical imaging research, where preserving lesion morphology is critical for diagnostic performance (Esteva et al., 2017).

Standardizing the image size also improves training stability and efficiency, as the model processes uniformly shaped inputs and doesn't need to adapt dynamically to different resolutions or shapes. In summary, these preprocessing steps help ensure that the most diagnostically useful information in the image is preserved and enhanced, while also adapting the data for compatibility with widely used model architectures.

3.2. CLAHE

Applying Contrast Limited Adaptive Histogram Equalization (CLAHE), along with padding and resizing images to 224x224 pixels, serves important purposes in preparing medical images.

CLAHE is a preprocessing technique used to improve the local contrast of an image. In the context of skin lesion classification, where subtle differences in color and texture are critical for diagnosis, enhancing contrast can help make those fine-grained features more visible. Traditional histogram equalization tends to over-amplify noise in homogeneous regions, which can be problematic in medical images. CLAHE, on the other hand, limits the amplification and works adaptively in small regions of the image, ensuring that areas like the lesion border or texture are brought into clearer focus without distorting the surrounding skin tone. This makes it easier for the model to learn relevant features, especially when the contrast between healthy and abnormal tissue is low.

CLAHE has been successfully applied in medical imaging tasks to enhance the visibility of diagnostically important features without amplifying noise, including dermatological image enhancement (Zuiderveld, 1994; Reza, 2004).

3.3. Hair removal with DullRazor

Hair removal was explored using the DullRazor algorithm. Although the method effectively detected and removed hairs, it introduced substantial blurring to the images. Models trained with this preprocessing step exhibited a noticeable decline in performance, suggesting that the negative impact of the blurring outweighed the benefits of hair removal. Consequently, **hair removal was excluded from the final preprocessing pipeline.**

Although hair removal preprocessing is recommended in dermoscopic image analysis (Abbas et al., 2013), it must be applied carefully to avoid introducing artifacts that degrade model performance.

3.4. Data Augmentation

Data augmentation is a widely used technique to improve model generalization, especially in medical imaging where datasets are often small (Shorten and Khoshgoftaar, 2019; Perez and Wang, 2017). In the context of skin lesion image classification, applying various data augmentation techniques during training is crucial to improving the model's generalizability. The data augmentation steps that were performed are as follows:

- Horizontal flip
- Color jitter
- Random affine transformations
- Vertical flip
- Adjusting sharpness randomly
- Random perspective shifts

3.4.1. Horizontal Flip

One of the most fundamental transformations is the random horizontal flip, which introduces variation in the orientation of lesions. Since lesions can appear anywhere on the body and from any angle, their horizontal positioning should not influence the model's decision. Flipping images horizontally encourages the model to focus on intrinsic features of the lesion rather than their orientation or placement within the frame.

3.4.2. Color Jitter

Another valuable augmentation is color jitter, which randomly adjusts brightness, contrast, saturation, and hue. This simulates a wide range of lighting conditions and device-specific color variations that might occur when collecting data in real-world clinical environments. Such variability is especially important in dermatology, where skin tone and lighting can heavily influence the appearance of lesions. By exposing the model to this variability, it becomes more resilient and less dependent on lighting conditions or fixed color patterns.

3.4.3. Random Affine Transformations

Random affine transformations, which include slight rotations, translations, scaling, and shears, help introduce spatial diversity to the dataset. Lesions may be captured at different angles, zoom levels, or positions within an image, and affine transformations prevent the model from becoming overly dependent on these spatial characteristics. Instead, the model learns to identify features that are invariant to such transformations—such as texture, asymmetry, or color distribution.

3.4.4. Random Vertical Flipping

Similarly, random vertical flipping can be useful, albeit less common than horizontal flipping, as it further diversifies the dataset's spatial representations. While vertical flips might be less frequent in natural imaging scenarios, they can still help in creating a model that is insensitive to unnatural orientation shifts, which could arise from camera positioning or patient movement.

3.4.5. Adjusting Sharpness Randomly

Another important transformation is adjusting sharpness randomly, which simulates the effect of varying focus levels. Not every medical image is taken under ideal circumstances—there may be slight blurriness due to motion or improper focus. By modifying sharpness, the model learns to be less sensitive to ultra-sharp, high-resolution features that might not always be present in real-world data, and instead focuses on more generalizable lesion characteristics.

3.4.6. Random Perspective Shifts

Finally, random perspective shifts simulate slight 3D distortions that occur when images are captured from different angles. This is particularly helpful in skin imaging, where a lesion photographed off-center or at an angle may appear distorted. Introducing perspective shifts during training ensures that the model does not overfit to front-facing, centered images and is better equipped to handle natural distortions or off-axis views.

Together, these augmentations serve an important role in ensuring that the model is not only accurate on the training data but also reliable when deployed in diverse and uncontrolled real-world environments. They collectively push the model to focus on the medical content of the image—the lesion itself—rather than on superficial or environment-specific artifacts.

4. Modeling approach

We use the following three architectures for our classification model:

1. MobileNetV3 Large
2. ShuffleNetV2
3. SqueezeNet1.1

Out of the three, **only ShuffleNetV2 and MobileNetV3 Large were used in the final ensemble model.** Because of the comparatively weak performance of **SqueezeNet1.1**, it was **excluded**.

The above-mentioned architectures were chosen because of their relative compactness, and that allows us to train quicker, thus enabling more frequent experimentation and iteration. The use of lightweight CNNs has been validated in prior research for balancing model performance and computational efficiency, especially for real-time or portable medical imaging applications (Howard et al., 2019; Ma et al., 2018; Iandola et al., 2016). The models are described below.

4.1. MobileNetV3

MobileNetV3 (Howard et al., 2019) is part of the MobileNet family of networks designed for mobile and embedded devices. It is optimized for both accuracy and efficiency by using a combination of depthwise separable convolutions and lightweight attention mechanisms (like the Squeeze-and-Excitation block). MobileNetV3 comes in two variants: MobileNetV3-Large and MobileNetV3-Small, with the large variant being optimized for better accuracy and the small variant for lower latency and fewer computations. The architecture uses a novel "h-swish" activation function and includes optimizations based on platform-specific searches.

4.2. ShuffleNetV2

ShuffleNetV2 (Ma et al., 2018) is an improved version of the ShuffleNet architecture, designed to be even more efficient than its predecessor. The key feature of ShuffleNetV2 is its use of channel shuffle operations, which reduce the computational cost of convolutional layers while maintaining performance. By improving the design of depthwise separable convolutions and introducing lightweight block structures, ShuffleNetV2 achieves faster inference speeds and higher efficiency compared to other networks with similar performance. The architecture includes a carefully designed building block that balances speed and accuracy, making it particularly suitable for mobile and edge devices.

4.3. SqueezeNet

SqueezeNet (Iandola et al., 2016) is a lightweight convolutional neural network (CNN) designed to achieve AlexNet-level accuracy with fewer parameters, making it efficient for use in resource-constrained environments. The key innovation of SqueezeNet is the use of "fire modules," which consist of a squeeze layer (with 1x1 convolutions) followed by an expand layer (with both 1x1 and 3x3 convolutions). This structure drastically reduces the number of parameters while maintaining high accuracy. SqueezeNet 1.1 improves upon the original SqueezeNet by further reducing the model size, making it even more efficient for tasks where memory and computation power are limited, such as mobile and embedded devices. This network is well-suited for tasks where model size and speed are more critical than absolute accuracy.

5. Training approach

5.1. Stratified k-Fold Cross Validation

In data preparation, we format the training dataset in k-fold cross validation form where we firstly take a test set which will be the same for all folds (10% of the entire dataset), and then the remaining dataset (90%) is split into folds, each fold having a distinct training (72% of the entire dataset) and validation (18% of the dataset) datasets. Stratified splitting ensures we have a fair distribution of classes in all folds.

5.2. Loss Function Choice

For this task, because it is unbalanced, we’ve applied two options:

1. Cross entropy loss, with class balancing
2. Focal loss

The final choice for the loss function was Focal loss.

5.2.1. Cross-entropy

Cross-entropy loss is the standard choice for multi-class or binary classification problems. However, when the data is imbalanced—say, significantly more benign than malignant samples—this loss tends to favor the majority class, leading to poor performance on underrepresented cases. To address this, the training dataset is rebalanced using a combination of oversampling the minority class and undersampling the majority class. This adjustment helps ensure the model sees enough examples from both classes and learns to treat them with equal importance. Class-balanced cross-entropy can also involve directly weighting the loss function, assigning higher penalties to misclassifications from the minority class. This helps guide the optimization process to give due attention to difficult or rare examples, which is essential in a medical setting where false negatives (e.g., missing a malignant lesion) are far more costly than false positives.

5.2.2. Focal loss

Focal loss, on the other hand, tackles class imbalance not by reweighting the dataset, but by modifying the loss function itself. It has proven particularly effective in medical image analysis tasks with strong class imbalance, improving detection of rare pathologies by focusing on difficult examples (Lin et al., 2017). Originally introduced for dense object detection tasks, focal loss adds a modulating factor to the standard cross-entropy loss, which down-weights well-classified examples and focuses the model's attention on harder, misclassified examples. This makes it particularly effective when there is a large class imbalance and many examples from the majority class are easy to classify, while the minority class contains harder, more subtle patterns. By reducing the influence of "easy" examples, focal loss ensures that the model spends more capacity learning the difficult cases, improving sensitivity on underrepresented classes.

5.3. Hyperparameter Tuning

We tune the following hyperparameters to find the best convergence on training/validation datasets:

1. Batch size
2. Learning rate
3. Dropout (L2 regularization)

5.3.1. Batch Size

The batch size determines how many samples the model sees before it updates its internal weights during training. Smaller batch sizes can lead to noisier but more frequent updates, which may help the model escape local minima and generalize better, especially on smaller datasets. However, they can also make training slower and more unstable. Larger batches, on the other hand, provide more stable gradient estimates and can speed up training with GPU acceleration, but they may lead to poor generalization and overfitting on imbalanced data. Tuning this hyperparameter is crucial for finding the right balance between stability, speed, and generalization.

5.3.2. Learning Rate

The learning rate controls the size of the steps taken during optimization. If the learning rate is too high, the model may overshoot optimal solutions, resulting in divergence or unstable training. If it's too low, training can become slow and might get stuck in suboptimal minima. Especially in transfer learning or fine-tuning pre-trained models—as is often done in medical image tasks—setting the learning rate appropriately is essential to adapt the model without destroying pre-learned features. Often, a learning rate scheduler or cyclical learning rate strategy is used in combination with tuning to improve convergence further.

5.3.3. Regularization

Dropout, or more broadly, L2 regularization, is a way to reduce overfitting, which is particularly important in medical applications where datasets can be small and overparameterized models can easily memorize training data. Dropout randomly disables a portion of neurons during training, encouraging the network to learn more robust and generalized features. L2 regularization penalizes large weights, effectively preventing the model from becoming too complex. Tuning these values ensures that the model is neither under-regularized (leading to overfitting) nor over-regularized (leading to underfitting).

5.4. Early Stopping

Early stopping is done using a patience of 10 epochs. If the validation loss doesn't go below the previous lowest score, the training will be stopped. This is done to reduce the computational cost of training the full desired number of epochs, if there is no convergence for a certain amount of time. It also helps reduce overfitting by not making training last long enough for the validation loss to theoretically diverge from the training loss (which could keep reducing indefinitely).

5.5. Evaluation Metrics

The metrics that are used for evaluating performance on training, validation and testing datasets are the following:

1. Accuracy
2. Precision
3. Recall (Sensitivity)
4. F1 Score

5. Specificity
6. Area Under the ROC Curve
7. Area Under the Precision-Recall Curve
8. Mean Loss
9. Confusion Matrix
10. Best Area Under the ROC Curve

5.5.1. Accuracy

Accuracy is the most intuitive metric—it calculates the proportion of correctly classified samples over the total number of samples. While it can give a quick snapshot of performance, it becomes less informative in imbalanced datasets. For example, if only 10% of the images are malignant and the model classifies all images as benign, it would still achieve 90% accuracy, despite failing entirely at detecting malignancies.

5.5.2. Precision

Precision measures how many of the samples predicted as positive (e.g., malignant) are actually positive. It's particularly important when the cost of false positives is high. In the context of melanoma detection, a false positive may cause unnecessary anxiety, further testing, or even biopsies. High precision means the model makes fewer of these costly mistakes.

5.5.3. Recall

Recall (Sensitivity), on the other hand, tells us how well the model identifies all the actual positives. A high recall means that most malignant cases are being detected. In medical applications, this is often more important than precision, because missing a melanoma case (false negative) could delay treatment and worsen outcomes.

5.5.4. F1 Score

F1 Score balances precision and recall by computing their harmonic mean. This is particularly useful when we care about both false positives and false negatives, and when the dataset is imbalanced. A high F1 score indicates that the model does well in both catching positive cases and avoiding false alarms.

5.5.5. Specificity

Specificity is the true negative rate—it tells us how well the model identifies benign cases. In other words, it’s the counterpart to recall but for the negative class. It’s important to ensure that benign lesions aren’t mistakenly flagged as malignant, which could lead to overtreatment or anxiety.

5.5.6. AUROC

AUROC (Area Under the Receiver Operating Characteristic Curve) is a powerful metric that captures the trade-off between sensitivity (recall) and specificity across all classification thresholds. A model with AUROC = 1 is perfectly able to distinguish between the classes, while a score of 0.5 indicates random guessing. AUROC is particularly useful when the dataset is imbalanced, as it’s less affected by class prevalence.

5.5.7. AUPRC

AUPRC (Area Under the Precision-Recall Curve) is often more informative than AUROC in the context of imbalanced data. It focuses on how well the model ranks true positives among all positive predictions and is especially helpful when we’re primarily concerned with the minority class—like malignant cases in skin lesion datasets. This metric helps us evaluate how effectively the model isolates malignant samples from the bulk of benign ones.

5.5.8. Mean Loss

Mean Loss reflects how well the model is learning during training. It’s a direct measurement from the loss function (e.g., cross-entropy or focal loss), averaged per epoch. A decreasing training and validation loss over time usually signals better fitting to the training data. Tracking training and validation loss allows us to identify underfitting or overfitting trends early on.

5.5.9. Confusion Matrix

Confusion Matrix is a 2x2 table (in binary classification) that gives a detailed breakdown of predictions:

- True Positives (correctly identified malignant),
- False Positives (benign predicted as malignant),
- True Negatives (correctly identified benign),

- False Negatives (malignant predicted as benign).

This matrix gives us a concrete view of how and where the model is making mistakes, which can guide further improvements in data balancing, model architecture, or training procedure.

5.5.10. ROC AUC Best

ROC AUC Best is a checkpointing reference that keeps track of the highest AUROC score achieved on the validation set during training. This is especially useful for model selection—saving the model state that performs best in distinguishing between benign and malignant cases ensures we don’t deploy an underperforming model from an arbitrary epoch.

6. Inferencing

During inference, instead of relying on a single model, we're using an ensemble—a group of the best-performing models identified during training. Each model makes a prediction, typically in the form of class probabilities. These predictions are then aggregated using soft voting, which means we average the predicted probabilities across the models and take the class with the highest mean probability as the final output. This technique is powerful because it reduces model variance and mitigates the weaknesses of individual models. Since each model may learn slightly different patterns or generalize differently, ensembling combines their strengths, leading to improved performance and more reliable predictions—especially valuable in medical tasks where robustness is crucial.

The same preprocessing steps are applied (CLAHE, padding, resizing).

Once the best-performing models have been selected through training and validation, they are exported to the ONNX (Open Neural Network Exchange) format. ONNX is an open standard designed to make models portable and interoperable across different frameworks. This allows models trained in environments like PyTorch or TensorFlow to be run independently using the ONNX Runtime, a fast and flexible inference engine. By using ONNX, the models can be deployed across various platforms (Windows, Linux, macOS, low-resource edge devices). This portability ensures that the deployment process is smooth and adaptable.

In addition to cross-platform compatibility, ONNX Runtime is also optimized for performance. It supports hardware acceleration, allowing the inference process to take advantage of GPUs, CPU vectorization, and even specialized AI hardware. This optimization leads to faster and more efficient inference.

Another major benefit of exporting models to ONNX is the resulting lightweight inference engine. Unlike traditional setups that depend on the full training framework (such as PyTorch, the CUDA toolkit, and a variety of large dependencies), the ONNX inference pipeline requires only a minimal set of libraries. These include ONNX Runtime itself, NumPy for basic tensor operations, and lightweight preprocessing tools like OpenCV and PIL for image loading and formatting. This significantly reduces the computational and memory overhead, making it easier to deploy models in production

7. Skin Tone Estimation

To estimate skin tone for each image, the following procedure was applied:

1. **Lesion segmentation** was performed using a pretrained model to isolate the lesion from the surrounding skin.
2. **ITA (Individual Typology Angle)** values were calculated based on pixel values outside the segmented lesion.
3. Each image was then assigned to one of six skin tone categories based on its ITA score: Very Light, Light, Intermediate, Tan, Brown, and Dark.

These ITA-based skin tones were subsequently mapped to Fitzpatrick skin types as follows:

- Very Light → Type I
- Light → Type II
- Intermediate → Type III
- Tan → Type IV
- Brown → Type V
- Dark → Type VI

This approach is robust to minor segmentation errors, as it only requires a reasonable estimate of the background skin tone to derive meaningful ITA values.

A more detailed explanation and the implementation of the skin tone estimation methods can be found in the skin_tone_exploration.ipynb notebook on the drive link.

8. Fairness Metrics

The metrics used to estimate the fairness of our model are as follows:

- Disaggregated Performance Metrics
- Demographic Parity
- Accuracy Parity
- Equality of Odds
- Calibration Across Groups
- Expected Calibration Error

These metrics allow us to quantify both the overall effectiveness of the model and its fairness in treating individuals with varying skin tones equitably. Recent studies (Groh et al., 2022) have highlighted the importance of ensuring that melanoma detection models perform equitably across skin tones, motivating the fairness metrics used in this project. A more detailed explanation and the implementation of the fairness analysis can be found in the fairness_exploration.ipynb notebook on the drive link.

8.1. Disaggregated performance

Disaggregated performance metrics refer to the basic performance indicators (accuracy, precision, recall, F1 score, and AUC) computed separately for each skin tone group. This allows us to assess how model performance varies across different Fitzpatrick types, rather than relying solely on overall metrics.

8.2. Demographic parity

Demographic parity evaluates whether a model treats different demographic groups equally - regardless of actual outcomes. In simple terms, it asks the question: Does the model predict the positive class (melanoma) at the same rate for different groups (skin types)?

8.3. Accuracy Parity

Accuracy Parity is a fairness metric that checks whether a model is equally accurate across different demographic groups (Fitzpatrick skin types). It asks: Is the overall classification accuracy consistent across groups?

8.4. Equality of Odds

Equality of Odds is a fairness metric that checks whether a model has equal error rates across demographic groups (Fitzpatrick skin types). It asks: Are both the True Positive Rate (sensitivity/recall) and False Positive Rate (fall-out) consistent across groups?

This ensures that the model:

- Detects disease equally well across all skin tones (equal TPR)
- Does not overpredict disease for any group (equal FPR)

8.5. Calibration Across Groups

Calibration Across Groups evaluates whether a model's predicted probabilities are equally reliable across different demographic groups. It asks: When the model predicts a probability of 70%, does that reflect a 70% actual chance of melanoma — for every group?

Calibration is especially important in disease detection because:

- Doctors may use the predicted probability to decide whether to biopsy a lesion.
- A poorly calibrated model may overestimate risk for one group and underestimate it for another, leading to unequal treatment.

The model is perfectly calibrated if the points for each group lie on the diagonal line

8.6. Expected Calibration Error

ECE stands for Expected Calibration Error — it is a metric that tells you how well a model's predicted probabilities match the actual outcomes. In other words: When your model says "there's a 70% chance this lesion is melanoma," is it actually right 70% of the time?

ECE quantifies the average mismatch between what your model thinks is true (its confidence) and what's actually true (the ground truth), across all probability levels.

9. References

- Abbas, Q., Celebi, M. E., & García, I. F. (2013). Hair removal methods for dermoscopy images: A review. *Biomedical Signal Processing and Control*, 6(4), 395–404. https://doi.org/10.1016/j.bspc.2011.11.003
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115–118. https://doi.org/10.1038/nature21056
- Groh, M., Harris, C., Soenksen, L. R., Lau, F., Han, R., Kim, A. S., Abid, A., Wu, N., Esteva, A., Yim, J., & Zou, J. (2022). Evaluating deep learning models for skin cancer diagnosis across skin tones. *Nature Medicine*, 28(6), 1312–1317. https://doi.org/10.1038/s41591-022-01772-4
- Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1314–1324). https://doi.org/10.1109/ICCV.2019.00140
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5MB model size. *arXiv preprint* arXiv:1602.07360. https://arxiv.org/abs/1602.07360
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.
  Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 2980–2988). https://doi.org/10.1109/ICCV.2017.324
- Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 116–131). https://doi.org/10.1007/978-3-030-01264-9_8
- Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint* arXiv:1712.04621. https://arxiv.org/abs/1712.04621
- Reza, A. M. (2004). Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 38(1), 35–44. https://doi.org/10.1023/B:VLSI.0000028532.52210.1f
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60. https://doi.org/10.1186/s40537-019-0197-0

- Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In P. S. Heckbert (Ed.), *Graphics Gems IV* (pp. 474–485). Academic Press. https://doi.org/10.1016/B978-0-12-336155-4.50061-6
- Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The 'K' in k-fold cross validation. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (pp. 441–446)