

Learning Orientation Cues in Natural Images with Embedded Machine Learning

Peirong Jiang

*University of Illinois at Urbana-Champaign, Champaign, USA
oscarj2@illinois.edu*

Abstract. Computer vision enables intelligent agents to extract meaningful information from images and make decisions based on visual inputs. Efficiency is the core demand within a limited computational resource. This research investigates how visual information can be processed by a lightweight vision system using embedded machine learning techniques. Using the Edge Impulse platform to simulate the vision module of an AI agent, by referring to Occlusion heatmap to evaluate the ability of lightweight Convolutional Neural Networks model when detecting horizontal image transformations under different visual cues. The model is trained on a small dataset of natural images under three experimental settings with consistent and randomized directional cues. Experimental results show that it is difficult to detect the flipping of natural images due to the visual symmetry. Whereas, with the introduction of consistent directional cues, the model is able to learn orientation differences more effectively. Occlusion heatmap further explains that the model fails to learn a position-invariant orientation representation and instead exploits location-dependent visual patterns.

Keywords: Computer Vision, Orientation Detection, CNN, Embedded Systems, Occlusion Heatmap

1. Introduction

Recent AI development increasingly adopts agent-oriented systems, especially with Visual Language Models and Large Language Models [1]. AI agents typically consist of modules for perception, planning, tool use, and decision-making [2], among which perception is crucial for understanding the world and supporting decisions [3]. Early vision methods relied on handcrafted features like SIFT and HOG, but these struggle with large-scale, complex visual environments [4]. Convolutional Neural Networks (CNNs) have become dominant because they learn hierarchical features directly from pixels and capture spatial structure via local receptive fields [5]. CNNs are widely used in demanding image analysis. Yu et al. reviewed architectures like AlexNet, GoogLeNet, and ResNet in medical imaging, finding that automatic feature learning significantly improves classification, detection, and segmentation accuracy [6]. Tan and Le proposed EfficientNet, which uses compound scaling to achieve high performance with fewer parameters [7]. These advances confirm CNNs' effectiveness in complex visual tasks.

However, although CNN models perform well at image recognition, standard CNNs are weak at handling spatial transformations such as rotation and scale, and may not inherently achieve robust

invariance. This limitation is further amplified in lightweight CNN models, which tend to exhibit reduced robustness and adaptability under resource constraints. This study aims to investigate whether a lightweight CNN model can learn to detect horizontal flips in natural images within an embedded vision environment. To examine the nature of the learned representations, the study systematically compare two conditions: natural images without explicit cues, and images with a directional cue ("L") placed either at a fixed position or at random positions. Occlusion heatmaps are further employed to visualize the model's decision-making process and assess whether it relies on position-invariant orientation features or location-dependent patterns. Based on this design, the study also provides a practical pipeline for robustness testing of embedded vision models.

2. Literature review

Researchers have gradually demonstrated that CNNs are highly suitable for image processing. Cywińska et al. proposed DeepOrientation, a method that uses a CNN to estimate fringe pattern orientation maps [8]. They concluded that CNNs can handle complex local direction estimation problems and outperform traditional methods in terms of robustness and automation. This demonstrates that CNNs are capable of learning directional information directly from image data.

CNNs have also shown strong performance in tasks that require fine-grained spatial understanding. Weng and Zhu reported that CNN-based systems achieved significant success in biomedical image segmentation, including brain MRI and chest X-ray analysis [9]. They found that CNNs can learn hierarchical features and perform pixel-level classification, making them highly suitable for complex image analysis. Moreover, Upreti et al. conducted a survey on the application of Convolutional Neural Networks in medical image understanding [10]. The study shows that CNNs can automatically learn feature representations directly from the original input images, unlike traditional image classification methods that rely heavily on manual feature analysis. This advantage enables CNNs to be efficient and effective in learning orientation-related cues and detecting flipped images.

However, the existing studies mainly focus on CNN models in specific tasks, such as medical image analysis and orientation estimation. There is limited attention to exploring the behavior of CNN models under varying visual conditions. Whether CNNs can learn robust, position-invariant representations or are influenced by location-dependent patterns is unclear. Moreover, these prior studies were conducted on standard or large-scale CNN architectures, while the behavior and limitations of lightweight models in resource-constrained environments remain underexplored.

3. Methodology

3.1. Model architecture

This study used the Edge Impulse platform, which provides built-in tools for developing embedded machine learning systems. Because of the relatively small data scale, the study selected MobileNetV2 as the base model. MobileNetV2 features depthwise convolutions and inverted residual blocks, which reduce computational complexity and the number of parameters. The training process followed the default configuration provided by Edge Impulse, without additional layer-freezing or layer-wise learning rate adjustments. Compared with traditional deep models, MobileNetV2 is less prone to overfitting under small-sample conditions, and therefore performs well in image recognition when deployed on an embedded system, despite its lightweight design.

Figure 1 illustrates the overall architecture. The input layer receives 224×224 pixel RGB images. A MobileNetV2 layer with a width multiplier of 0.35 extracts visual features while retaining only 35% of the channels in each layer, further reducing model size. The network then passes the extracted features to a fully connected layer with 16 neurons, followed by a dropout layer with a rate of 0.1 for regularization. The output layer contains two neurons with a SoftMax activation function to perform binary classification between original and flipped images.

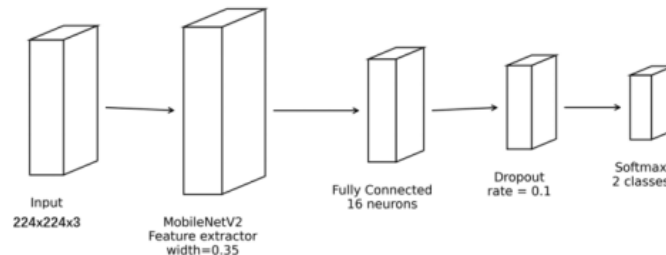


Figure 1. Convolutional neural network structure

3.2. Dataset

This study sourced images from Unsplash.com, a publicly accessible platform known for its diverse, high-quality images [11]. The study selected cats as target objects because their variability in pose, texture, and orientation supports learning robust and generalizable visual features. The study divided the collected images into training and testing sets at an 8:2 ratio, yielding 105 training images and 27 testing images. The study then applied the Python built-in function `ImageOps.mirror()` to horizontally flip all images. The flipped versions received the label "flipped," while the original images received the label "normal." During preprocessing, the Edge Impulse platform automatically normalizes and resizes images to ensure consistent input dimensions for the CNN model.

To introduce explicit directional cues, the study added a white letter "L" to the images in two configurations:

Fixed-position cue (Figure 2): The letter "L" was placed at the top-left corner, with its size set to 12% of the image width. Each image was then mirrored, causing the letter to appear on the opposite side in the flipped version.

Random-position cue (Figure 3): The letter "L" of the same size was placed at a random position in each original image, and then the images were mirrored to obtain the flipped set.



Figure 2. Examples of images with a fixed-position directional cue



Figure 3. Random-position cue: letter "L" at a random location

3.3. Training process

All three experimental stages used the same training configuration to ensure comparable results. The study set the number of training epochs to 30, with a learning rate of 0.005 to enable relatively fast convergence while maintaining training stability. Given the balanced nature of the binary classification task, the study randomly sampled 20% of the training data (without stratification) to serve as the validation set.

The three stages differed only in the type of input data. The first stage used natural images, allowing the CNN to learn orientation-related patterns from unaltered visual scenes. The second stage introduced images with a fixed-position letter "L" as a strong directional cue. The third stage used images with the letter "L" placed at random positions to test whether the model relied on fixed spatial features or could generalize to position-invariant orientation patterns. The study applied no additional data augmentation during training.

3.4. Generate the Occlusion heatmap

The Occlusion heatmap was generated in Python using the trained model configuration in TensorFlow Lite format. Set the occlusion patch size to 12×12 pixels to balance spatial resolution and contextual coverage. The stride was set to 6 pixels, resulting in 50% overlap between adjacent patches. By evaluating each pixel multiple times, a more robust importance estimate can be obtained through averaging.

The importance of each pixel is computed as:

$$I(x,y)=S_{base}-S_{occ}(x,y) \quad (1)$$

Where S_{base} is the predicted confidence of the original image, and $S_{occ}(x,y)$ is the prediction confidence after occluding the region (x,y) , higher values of $I(x,y)$ indicate a greater contribution of that region to the model's prediction. Only regions with $I(x,y)>0$ were retained, as they indicate a positive contribution to the model's prediction, and take the average of all importance value of pixels if covered by multiple overlapping patches. Finally, normalize the importance $I(x,y)$ between $[0,1]$ for visualization.

4. Result

4.1. Stage one: natural images

The first experiment was conducted using natural images. After testing the model, the model achieved a classification accuracy of 25.93%. The area under the ROC curve (AUC), the weighted average precision, recall, and F1 score were all 0.54. Table 1 shows only 18.5% images are correctly classified as the flipped image, while 37.0% are misclassified as normal images, and 44.4% were classified as uncertain. For normal images, 37.0% were correctly classified, while 33.3% were not, and 29.6% are uncertain. The F1 scores shown were 0.24 for flipped images and 0.39 for normal images, demonstrating that the model's performance was limited when no explicit directional cues were present in the images. The classification results are visualized using scatter plots to illustrate classification performance and the distribution of predicted classes. Figure 4 shows all the image classification distributions, where colors indicate correct and incorrect classifications for both normal and flipped categories. The overall distribution is scattered.

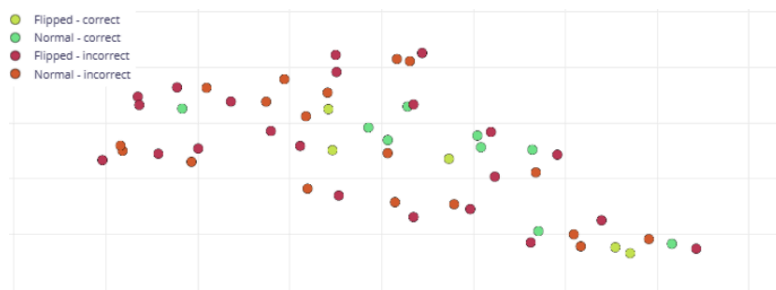


Figure 4. Scatter plot of classification results for Experiment 1

4.2. Stage two: images with orientation cue

This experiment was conducted using images with a white letter "L" added at the top-left corner as a directional cue. As shown in table 1, the overall accuracy was 61.54%, with the area under the ROC curve (AUC), weighted average precision, recall, and F1 score all at 0.67. For flipped images, 65.4% were correctly classified, 23.1% were misclassified as normal, and 11.5% were classified as uncertain. For normal images, 57.7% were correctly classified, 30.8% were misclassified as flipped, and 11.5% were uncertain. The per-class F1 scores were 0.64 for both flipped and normal images.

The classification results are visualized in Figure 5. Compared with Experiment 1, the scatter plot shows a clear shift: data points are clustered toward the left and right sides, indicating improved class separation when a consistent directional cue is present.

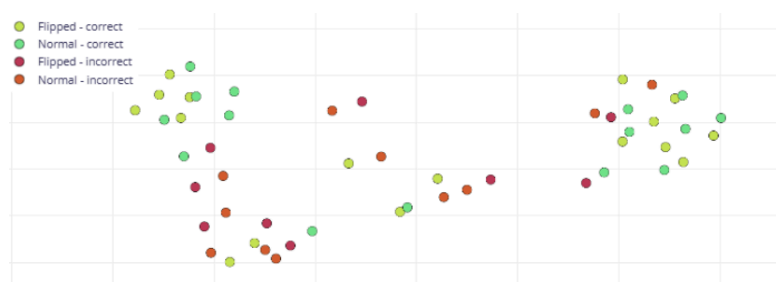


Figure 5. Scatter plot of classification results for Experiment 2

4.3. Stage three: images with random position of "L."

The position of the inserted orientation marker "L" was randomized across the image. The overall classification accuracy was 29.63%, and the area under the ROC curve (AUC), weighted average precision, recall, and F1 score were all 0.44. A high level of misclassification was observed between the two classes. For flipped images, 40.7% were correctly classified, 33.3% were misclassified as normal, and 25.9% were uncertain. For normal images, only 18.5% were correctly classified, while 59.3% were misclassified as flipped and 22.2% were uncertain. The F1 score for flipped images was 0.41, and for normal images it was 0.24.

The scatter plot for Experiment 3 (Figure 6) shows that although the data points are densely clustered around the center, they do not exhibit clear class separation.

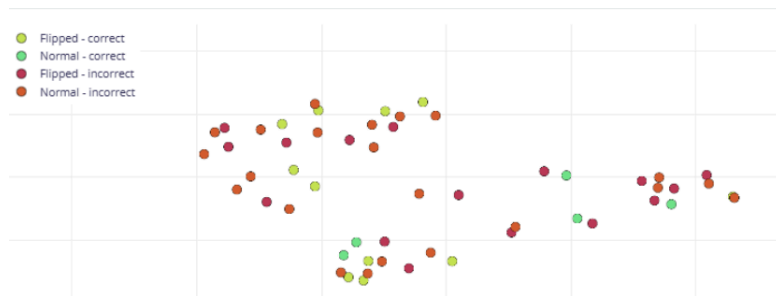


Figure 6. Scatter plot of classification results for Experiment 3

Table 1. Summary of classification performance across three experiments

Experiment	Accuracy	AUC	F1 (Flipped)	F1 (Normal)
No cue	25.93%	0.54	0.24	0.39
Fixed-position cue	61.54%	0.67	0.64	0.64
Random-position cue	29.63%	0.44	0.41	0.24

4.4. Occlusion heatmap



Figure 7. Occlusion heatmaps for Experiment 2. (a) Correctly classified normal image (confidence = 0.79). (b) Correctly classified flipped image (confidence = 0.99). (c) Incorrectly classified as flipped. Warmer colors indicate greater contribution to the model's prediction

Three cases were selected to represent different outcomes: a correctly classified normal image, a correctly classified flipped image, and an incorrect classification. Figure 7 shows the occlusion

heatmaps for these three cases. In (a), the model correctly predicts "normal" (0.79 confidence), with highlighted regions in the upper-right and lower-left corners, away from the inserted "L." In (b), the model correctly predicts "flipped" (0.99 confidence), and again the important regions do not align with the "L." In (c), the model mistakenly predicts "flipped" and focuses on a region near the "L," suggesting that attending to the marker alone does not guarantee correct classification. A Cochran's Q test comparing binary classification accuracy across the three experimental stages yielded $Q = 17.33$ ($p = 0.00017$), indicating statistically significant differences in performance across conditions.

5. Discussion

5.1. Findings

The comparison across the three experiments suggests that strong directional cues may improve the model's orientation classification performance. When the fixed-position letter "L" was introduced, the model showed clearer category separation and more balanced classification behavior compared with the natural-image condition. However, once the position of the letter "L" became random, model performance decreased substantially, indicating that the learned representation may depend more heavily on spatial correlations than on generalizable orientation features.

The occlusion heatmaps support this interpretation. Important regions did not consistently align with the letter "L," even when predictions were correct. In some cases, the model focused near the "L" yet still misclassified. One possible explanation is that when multiple visual features (e.g., "L", background textures, edge orientations, and spatial positions) are present in the images, the model will automatically conduct feature competition, focusing on the features that minimize the loss function. This reveals that the model does not consistently rely on the semantics of the inserted letter "L" to make predictions.

Overall, these findings highlight the importance of evaluating model robustness and interpretability, rather than relying solely on accuracy, especially in embedded vision systems where reliability under varied conditions is critical.

5.2. Limitations

This paper uses a lightweight CNN whose simple structure may not fully capture complex visual features. The small scale and limited diversity of the training and test sets may further restrict generalization, making results sensitive to specific data distributions. The large performance gap across experiments and the reappearance of class bias suggest the model relies on dataset-specific patterns rather than learning stable orientation features. Future research can build on this study by introducing a larger, more diverse dataset to enhance the model's robustness and generalization, and by using more complex model structures to strengthen its ability to identify key features.

6. Conclusion

This study investigates whether a lightweight embedded vision model can learn image orientation under different visual conditions. Introducing a fixed-position letter "L" as a directional cue improves classification accuracy from 25.93% to 61.54%, confirming that explicit cues aid orientation detection. However, when the cue position is randomized, accuracy drops sharply to 29.63%, indicating that the model fails to generalize across spatial variations and does not learn a

position-invariant orientation representation. Occlusion heatmap analysis further reveals that the model relies on location-dependent visual patterns rather than the semantics of the directional cue.

Several limitations should be noted. The lightweight CNN architecture used in this study may lack the capacity to learn robust, generalizable features, and the relatively small dataset limits the diversity of visual patterns the model encounters. Future work could employ more expressive model architectures and larger, more diverse datasets.

References

- [1] Piccialli, F., Chiaro, D., Sarwar, S., Cerciello, D., Qi, P., & Mele, V. (2025). AgentAI: A comprehensive survey on autonomous agents in distributed AI for industry 4.0. *Expert Systems with Applications*, 291, 128404.
- [2] Krishnan, N. (2025). Ai agents: Evolution, architecture, and real-world applications. *arXiv preprint arXiv: 2503.12687*.
- [3] Durante, Z., Huang, Q., Wake, N., Gong, R., Park, J. S., Sarkar, B., ... & Gao, J. (2024). Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv: 2401.03568*.
- [4] Trigka, M., & Dritsas, E. (2025). A comprehensive survey of machine learning techniques and models for object detection. *Sensors*, 25(1), 214.
- [5] Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 4712.
- [6] Yu, H., Yang, L. T., Zhang, Q., Armstrong, D., & Deen, M. J. (2021). Convolutional neural networks for medical image analysis: state-of-the-art, comparisons, improvement and perspectives. *Neurocomputing*, 444, 92-110.
- [7] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.
- [8] Cywińska, M., Rogalski, M., Brzeski, F., Patorski, K., & Trusiak, M. (2022). DeepOrientation: convolutional neural network for fringe pattern orientation map estimation. *Optics Express*, 30(23), 42283-42299.
- [9] Weng, W., & Zhu, X. (2021). INet: convolutional networks for biomedical image segmentation. *Ieee Access*, 9, 16591-16603.
- [10] Upreti, M., Pandey, C., Bist, A. S., Rawat, B., & Hardini, M. (2021). Convolutional neural networks in medical image understanding. *Aptisi Transactions on Technopreneurship (ATT)*, 3(2), 120-126.
- [11] Unsplash. (n.d.). Beautiful free Images & Pictures. Retrieved March 19, 2026, from <https://unsplash.com/>