frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 19 September 2025
DOI 10.3389/fcomp.2025.1644044

OPEN ACCESS

EDITED BY
Xiaohao Cai,
University of Southampton, United Kingdom

REVIEWED BY
Rui-Yang Ju,
National Taiwan University, Taiwan
Jiahong Zhang,
University of Chinese Academy of Sciences, China

*CORRESPONDENCE
Lukas Tuggener
lukas.tuggener@rwai.ch

RECEIVED 09 June 2025
ACCEPTED 13 August 2025
PUBLISHED 19 September 2025

CITATION
Tuggener L, Stadelmann T and Schmidhuber J (2025) Efficient rotation invariance in deep neural networks through artificial mental rotation. *Front. Comput. Sci.* 7:1644044. doi: 10.3389/fcomp.2025.1644044

Efficient rotation invariance in deep neural networks through artificial mental rotation

Lukas Tuggener[1,2]*, Thilo Stadelmann[1,3] and Jürgen Schmidhuber[2,4]

[1]ZHAW Centre for Artificial Intelligence, Winterthur, Switzerland, [2]Faculty of Informatics, University of Lugano, Lugano, Switzerland, [3]European Centre for Living Technology, Venice, Italy, [4]AI Initiative, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Humans and animals recognize objects irrespective of the beholder's point of view, which may drastically change their appearance. Artificial pattern recognizers strive to also achieve this, e.g., through translational invariance in convolutional neural networks (CNNs). However, CNNs and vision transformers (ViTs) both perform poorly on rotated inputs. Here we present AMR (artificial mental rotation), a method for dealing with in-plane rotations focusing on large datasets and architectural flexibility, our simple AMR implementation works with all common CNN and ViT architectures. We test it on randomly rotated versions of ImageNet, Stanford Cars, and Oxford Pet. With a top-1 error (averaged across datasets and architectures) of 0.743, AMR outperforms rotational data augmentation (average top-1 error of 0.626) by 19%. We also easily transfer a trained AMR module to a downstream task to improve the performance of a pre-trained semantic segmentation model on rotated CoCo from 32.7 to 55.2 IoU.

KEYWORDS

computer vision, mental rotation, CNN, transformer, in-plane rotations, bio-inspired, neural network architecture

1 Introduction

Natural vision systems in humans and animals are able to recognize objects irrespective of transformations such as rotations and translations as well as the observer's point of view, all of which can have a tremendous impact on the appearance of said object. This is a highly desirable property for all vision systems, especially if they are to be deployed in real-world settings that are characterized by significant scale and visual complexity (Stadelmann et al., 2018, 2019). CNNs (Fukushima et al., 1980, 1983; Waibel et al., 1989; Zhang et al., 1988) inherently integrate translational invariance into their design. Vision transformers (Vaswani et al., 2017; Dosovitskiy et al., 2021) [a.k.a. neural fast weight programmers (Schmidhuber, 1992; Schlag et al., 2021)] exhibit a level of translational robustness however, they are not fully translational invariant (Rojas-Gomez et al., 2024). For rotations, this is not the case, and both methods perform very poorly when facing inputs at an unusual angle (see Figure 1 and Engstrom et al., 2019; Xu et al., 2023). This can be exploited for adversarial attacks (Engstrom et al., 2019) and thus can cause serious issues in applications where rotated inputs are common.

A widely used approach is based on input data augmentation. The data is rotated at training time such that the model can learn all appearances of an object. This yields good results and can scale to any problem size. It is, however, an inefficient method, reducing the sample efficiency and consequently resulting in lower final performance with equal training time (see differences at angle zero in Figure 1).

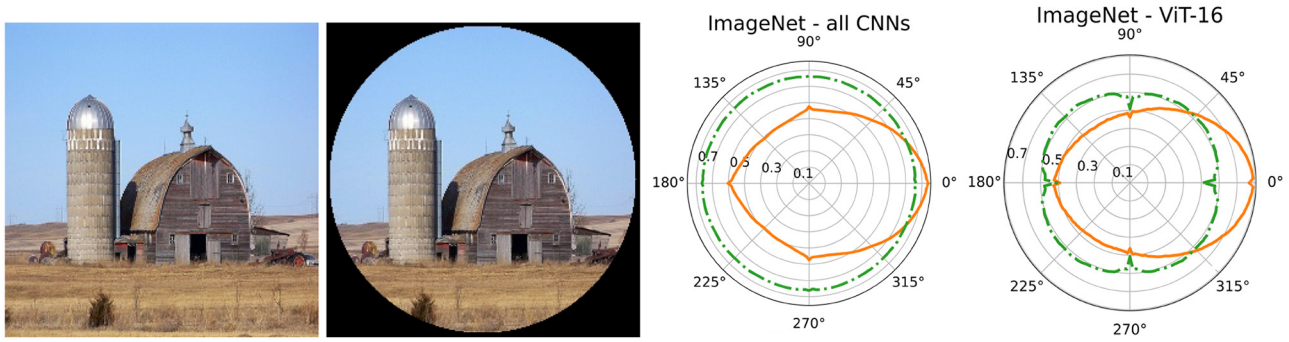

FIGURE 1
An example image from ImageNet (far left) and a version of the same image with its corners masked, to allow for non-obvious and reversible rotations (center left). Polar plots of ImageNet top-1 accuracies by angle averaged across CNN eight architectures (center right) and for VIT-16 (far right). The performances in solid orange are for models with standard training, the dash-dotted green lines represent training with rotational data augmentation. For training details see Chapter 3. Image taken from the ILSVRC2012 version of ImgeNet. https://www.image-net.org/challenges/LSVRC/2012/.

A straightforward approach to achieving rotation invariance focuses on building architectures that incorporate rotational invariance or sometimes equivariance directly into the neural network design (Cohen and Welling, 2016; Marcos et al., 2017; Dieleman et al., 2015; Cohen and Welling, 2022; Weiler and Cesa, 2019; Laptev et al., 2016; Worrall et al., 2017; Kaba et al., 2023; Mondal et al., 2023). However, the largest gains in model performance in the last years have been realized through systematic scaling up of model and training data size as well as training time (Tan and Le, 2019; Zhai et al., 2022). This trend led to growing model architectures and culminated in the inception of foundation models (Bommasani et al., 2021; Kirillov et al., 2023). These large and ever-evolving architectures are generally not rotation invariant (Mondal et al., 2023). We focus here on practical applicability and performance so we aim to leverage the power of these models directly, without change. To achieve this, a decoupled method facilitating the addition of rotation invariance is needed to solve the problem of degraded performance in the presence of in-plane rotations in an architecture-independent fashion. This demands methodological simplicity with low development overhead and fast execution w.r.t. runtime on top of excellent performance on various downstream vision tasks (e.g., classification or segmentation).

It is a long-standing conjecture in neuro-psychology that when humans try to identify an object, they mentally simulate rotations of that object to match it to an internal representation (i.e., they perform mental rotation). Shepard and Metzler (1971) were the first to formally study this phenomenon. They were able to show that the time human subjects need to determine whether pairs of 3D figures have the same shape grows linearly with the angle of rotation between the two objects. This strongly suggests that humans perform mental rotation; otherwise, the re-identification task would be completed in constant time across angles. This concept of mental rotation has been of inspiration to several computer vision methods (Ding and Taylor, 2014; Boominathan et al., 2016; Feng et al., 2019; Fang et al., 2020; Kaba et al., 2023; Mondal et al., 2023).

In this paper, we introduce the Artificial Mental Rotation (AMR) method that separates the finding of the angle of rotation of a given input from subsequently rotating it back to its canonical appearance before further processing, thus performing an artificial version of mental rotation. The problem of rotation estimation has been considered hard in the literature before Boominathan et al. (2016). However, it has the advantages that the angles can be found in a one-shot fashion and the underlying method of visual recognition itself does not have to be hardened against rotations, therefore all models (even trained ones) can be used in conjunction with an AMR module.

In short, our core contributions are: (a) We introduce a simple approach and corresponding self-supervised training method, AMR, for invariant processing of rotated images, (b) we present a simple neural network architecture that implements AMR and can be paired with all common CNNs and ViTs without alteration of the trained base model, (c) we extensively test the real-world merits of AMR on rotated versions of ImageNet, Stanford Cars, and Oxford Pet, and conclude that it significantly outperforms rotational data augmentation and generally shows excellent performance in practically relevant tasks, (d) we present AMR results on MNIST showing it performs competitively to existing methods, (e) we confirm the viability of AMR in a scenario where only portions of the test data are rotated, (f) we present comprehensive ablation studies proofing that our trained AMR modules work in practice on synthesized as well as physically rotated data, and (g) we show the easy transferability of a trained AMR module to another downstream vision task (in this case semantic segmentation), significantly increasing the performance of an existing model on rotated data.

2 Related work

An important early work is Spatiatial Transformer Networks (Jaderberg et al., 2015) which extends CNNs with the ability to learn spatial transformations (including rotations) for its feature maps. This architecture has since been tailored to specific equivariances (Esteves et al., 2018; Tai et al., 2019).

There are ongoing efforts to incorporate rotation invariance (or in some cases equivariance) directly into the architectures of deep neural networks, especially for CNNs. Dieleman et al. (2015) introduced a rotation invariant CNN system for galaxy morphology prediction that uses multiple rotated and cropped snippets of the same image as input. Cohen and Welling (2016) presented G-CNNs which are equivariant to a larger number of symmetries such as reflections or rotations. This is achieved by lifting the network's features to a desired symmetry group. The later work on steerable CNNs (Cohen and Welling, 2022; Weiler and Cesa, 2019) extended this work. Romero and Cordonnier (2021) presented group equivariant vision transformers by extending the symmetry group lifting concept to self-attention. Worrall et al. (2017) introduced H-Nets which replace regular CNN filters using circular harmonics. Marcos et al. (2017) have proposed to rotate the filters of a CNN and then apply spatial and orientation pooling to reduce and merge the resulting features. Laptev et al. (2016) introduced a TI-pooling, which allows to pool the CNN outputs for an arbitrary number of different angled versions of the same input to create an equivariant feature. These methods all entangle model architecture and equivariance properties.

Data augmentation (Baird, 1992) is very widely used to improve the robustness and generalizability of vision models (Simard et al., 2003). It can even be used to harden the model against adversarial attacks (Shafahi et al., 2019). Data augmentation has also been shown to be very effective for rotated inputs (Quiroga et al., 2020). Later work aims to improve the sample efficiency of data augmentation by directly learning object-specific invariances or transformation inference functions to inform the augmentation process (Miao et al., 2023; Immer et al., 2022; Allingham et al., 2024). While improving rotational stability do data augmentation approaches not tackle the issue on a fundamental level.

There have been previous attempts to leverage the concept of mental rotation for computer vision. Ding and Taylor (2014) trained a factored gated restricted Boltzmann machine to actively transform pairs of examples to be maximally similar in a feature space. Boominathan et al. (2016) train a shallow neural network to classify if an image is upright. They combine this with a Bayesian optimizer to find upright images and use this setup to improve image retrieval robustness. In the space of 3D vision, a mental rotation-based approach achieved state-of-the-art performance for rotated point cloud classification (Fang et al., 2020). In the representation learning community rotation prediction using CNNs has been leveraged as an additional, self-supervised, learning signal to train better representations (Feng et al., 2019). Kaba et al. (2023) achieve invariance via learned canonicalization functions. In a follow-up work (Mondal et al., 2023) adapt canonicalization functions to large pretrained models. While being unique, both of these works highly relate to this contribution, this relation is discussed in more detail in Section 7. The very recent TICNNs (Zhang et al., 2025) introduce a log-polar transformation inspired by the human retina and train a canonicalization function in LP space.

3 Artificial mental rotation

Our AMR approach requires three components. First, a base model (BM) is required, for which any common CNN or ViT (Dosovitskiy et al., 2021) architecture can be used. There is no need to modify the BM in any way, hence, the BM can generally be sourced in a fully (pre-)trained form. However, we do copy features out of the BM at various stages, in cases where this is not possible e.g., when using a BM hidden behind an API the AMR module has to be designed as a stand-alone network (this would be equivalent to Stem in Chapter 1). Additionally, it requires a rotation algorithm designed for images; here we use the method available in OpenCV (Bradski, 2000). The last necessary component is the AMR module itself, presented in this section. Due to their differing designs, CNNs and ViTs use slightly varying AMR modules.

3.1 AMR training

While training the AMR module, the BM is frozen such that its classification performance is not disturbed. For the training, we use datasets, like ImageNet, where the objects are typically shown in an upright position. Under this constraint, we can employ self-supervised training by randomly rotating the input images and asking the AMR module to recover the angle we previously applied.

3.2 AMR inference

AMR inference is performed in a three-step process: (1) The input's angle is classified by running it through the BM and the AMR module. (2) The input is rotated by the negative amount of the angle determined in step one. (3) The rotation-corrected input is processed by the BM. Step (3) is identical to AMR-free inference since the BM is frozen during AMR training and there is no information flow through the AMR module during this step. Therefore AMR could also be framed as a preprocessing method by reducing it to steps (1) and (2).

3.3 AMR module for CNNs

Our AMR module is designed as an add-on to a given BM (see Figure 2), so it can repurpose the features computed by the BM and only requires a small number of additional weights. Features are copied into the AMR module at five different BM stages. In the case of ResNe(X)ts (He et al., 2016; Xie et al., 2017) [a.k.a. Highway Nets with open gates Srivastava et al. (2015)] this happens directly after the stem and after each of the four ResNe(X)t stages. For EfficientNets (Tan and Le, 2019) we use the end of stages 2, 4, 6, and 8 as extraction points. When the copied features enter the AMR module they are first processed by a single 1×1 2D convolution to compress the feature depth. For all but the first AMR module stages, these features are then stacked with the output of the previous AMR module stage followed by another 1×1 2D convolution to half the feature depth of the stack. Only then is the data processed by a single ResNet block. After the last stage, we employ average pooling and a single fully connected layer with 360 outputs to create the angle prediction.

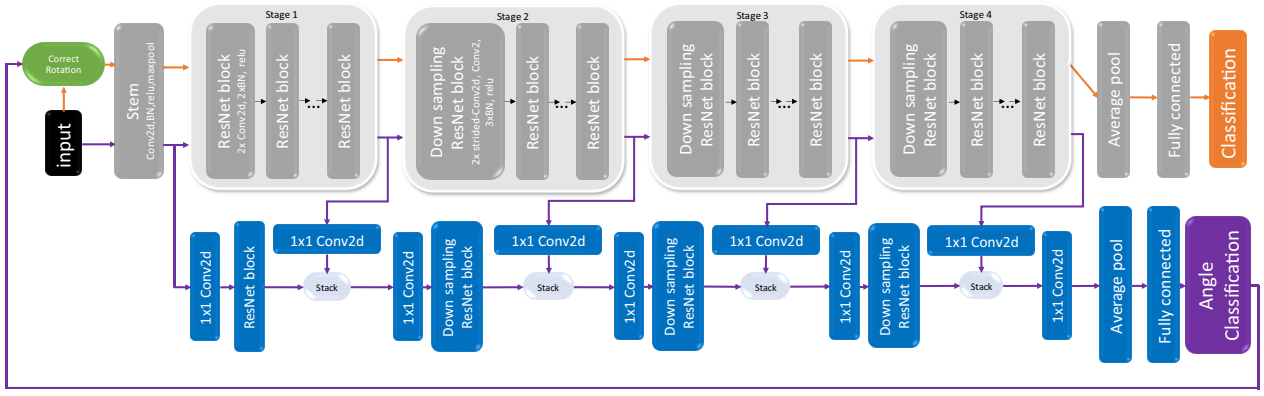

FIGURE 2
Architecture of our artificial mental rotation module for CNNs. The base CNN, in this case a ResNet, is shown in gray. The components of the AMR module are shown in blue. The information flow in stage 1 (angle classification) is purple while the information flow in stage 3 (image classification) is shown in orange.

3.4 AMR module for ViTs

The AMR module for ViTs functions very similarly to the one for CNNs. We again extract features at five different locations. For ViT-16-b these are after encoder blocks 1, 4, 7, and 12. Since there is no spatial downsampling in ViTs there is no advantage in processing the extracted features in stages. We, therefore, stack them all at once followed by a single 1×1 2D convolution. This stack is then processed by four ViT encoder modules. Lastly, we extract the same classification token that was used in the BM and apply a fully connected layer for the angle classification.

3.5 Motivation for add-on design

We opted to design our AMR module as an add-on to existing base networks because we conjecture that the features that have been trained for classification will also be at least partly useful for angle detection and the AMR module can profit from the training resources that have already been invested into the base network. We confirm this conjecture with an ablation study (see Section 1 in the Supplementary material). This design choice therefore, allows for an AMR module that consists of very few layers on its own. Thus, it can be trained very quickly and only adds a constant overhead of roughly 5 million parameter, resulting in 0.905 GFlops.

4 Experiments

We aim to showcase the merits of AMR on natural images. Therefore, we test it on ImageNet (ILSVRC 2012) (Russakovsky et al., 2015) and verify our results on Stanford Cars (Krause et al., 2013) and Oxford Pet (Parkhi et al., 2012), by employing rotated versions of the mentioned datasets. To ensure that artificial rotations are not obvious, we mask out the corners of all images such that a centered circle remains (see Figure 1 and the next Section for an ablation study ensuring the artificial rotations are not carrying any unwanted information). For a fair comparison between upright training (without data augmentation) and training with random rotations as input data augmentation (rotated training), we train all of our base models from scratch with this masking applied. For all of our training runs, we use image normalization based on dataset statistics. No further data augmentation is applied to keep the experiments as simple as possible (except, of course, input rotation for the rotated training models). To obtain representative results we replicate our experiments on a variety of base models. We use three different ResNets, three EfficentNets, and two ResNeXts for a total of eight CNN architectures. On ImageNet we also employ a vision transformer in the form of ViT-16b, which is unsuited for the other smaller datasets. For each upright trained base model, we train two AMR modules: One is trained for one-third of the base model's training time (in epochs) and the other one for one-twentieth.

4.1 Training details

For all base models, we use the implementations from the torchvision (TorchVision Maintainers and Contributors, 2016) Python package without any modifications. To enable optimal training speed our code is based on the ffcv library (Leclerc et al., 2022). All training details and links to code and trained model weights can be found in the Supplementary material.

4.2 Testing

We first evaluate the upright base models on upright data. We use these performances as the ceiling of what can be achieved on rotated data. Then we test the upright and rotated base models as well as the AMR-enhanced models for rotated performance by rotating the test set two degrees at a time and running a full evaluation for each angle. We present the resulting data visually in polar plots (see Figures 3, 4) as well as in table form (see Tables 1, 2) by averaging across angles.

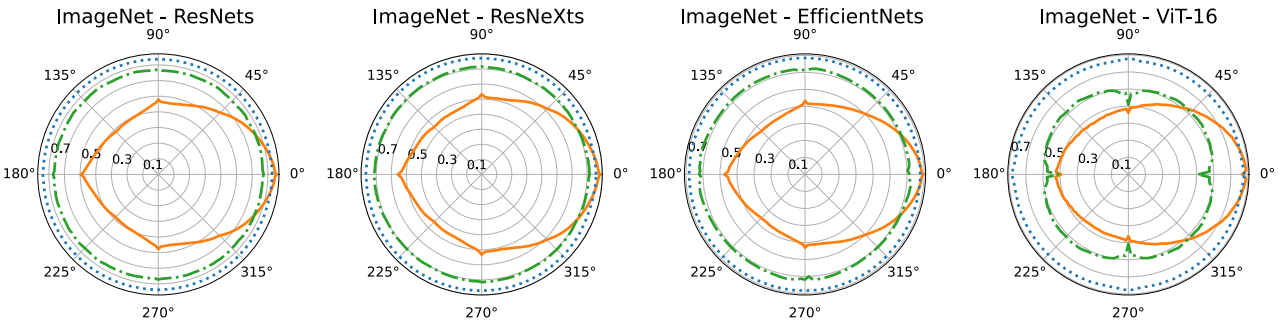

FIGURE 3
Polar plots of ImageNet top-1 accuracies by image rotation angle, averaged across architectures (except ViT). The performances of the upright base models are shown in solid orange, the rotated training base models are shown in dash-dotted green, and AMR performance (averaged across both epoch regimes) is shown in dotted blue lines.

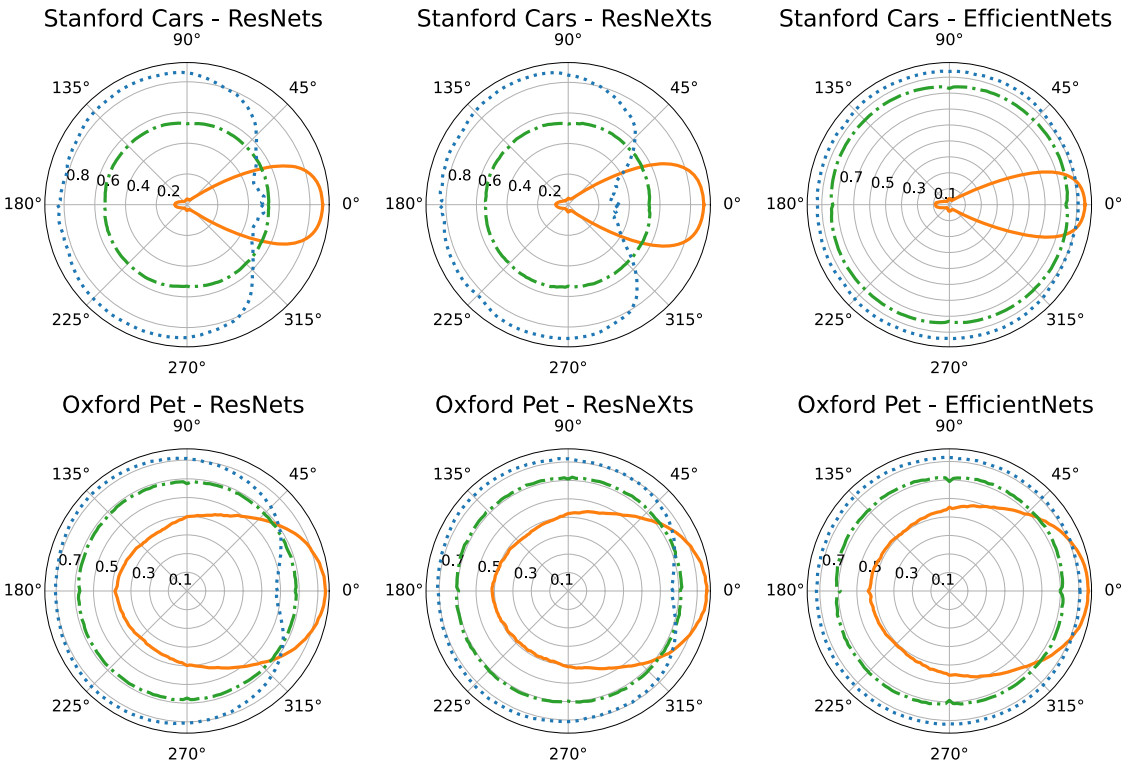

FIGURE 4
Polar plots of Stanford Cars (left column) and Oxford Pet (right column) top-1 accuracies analogous to the ones shown above for ImageNet (see Figure 3).

4.3 ImageNet

We train all of our base CNNs for 100 epochs on ImageNet, and the vision transformer is trained for 300 epochs, in accordance with the training recipes for the torchvision base models. We then train two AMR modules in conjunction with each upright trained base model, one for 33 epochs and the other for 5. Preliminary experiments have show that at 33 epochs the performance is saturated and that 5 epochs represents the best compromise between performance and training cost. Table 1 contains the top-1 accuracies on rotated data for all models. Additionally, the ceiling accuracy is also reported (upright data with upright trained model). As suspected, there is a steep drop in accuracy between upright and rotated testing for the upright-trained models, both for the CNNs as well as the ViT. On average only 69 percent of the ceiling performance (% ceil) is retained. The models which have been trained with random rotations fare much better, they achieve 87% ceil. It is noteworthy that the ViT only rises from 66 to 73 of the ceiling performance. This makes sense since ViTs tend to be less sample efficient compared to CNNs and therefore suffer more from the increased problem complexity caused by the random rotations. AMR-33 achieves 98% ceil, significantly outperforming

TABLE 1 ImageNet top-1 accuracies.

	Upright training			Rotated training		AMR 33		AMR 5	
Testing	up	rot	% ceil	rot	% ceil	rot	% ceil	rot	% ceil
ResNet-18	0.695	0.433	62	0.598	86	**0.676**	**97**	0.666	96
ResNet-50	0.768	0.537	70	0.673	88	**0.755**	**98**	0.746	97
ResNet-152	0.779	0.552	71	0.730	94	**0.767**	**98**	**0.760**	**98**
EfficientNet-b0	0.680	0.454	67	0.611	90	**0.666**	**98**	0.656	96
EfficientNet-b2	0.692	0.467	67	0.612	88	**0.678**	**98**	0.669	97
EfficientNet-b4	0.710	0.485	68	0.618	87	**0.696**	**98**	0.689	97
ResNext-50-32x4d	0.773	0.551	71	0.686	89	**0.761**	**98**	0.754	98
ResNext-101-32x8d	0.785	0.571	73	0.728	93	**0.772**	**98**	**0.766**	**98**
ViT-16b	0.691	0.459	66	0.503	73	**0.669**	**97**	0.664	96
Average	0.730	0.501	69	0.640	87	**0.716**	**98**	0.708	97

Upright testing (up) of the upright trained base model is assumed to be the performance ceiling (% ceil). Average (by angle) rotated accuracies (rot) are given for the upright and rotated trained base models as for AMR 33 epochs and AMR 5 epochs. The best performances are bold.

TABLE 2 Stanford cars and oxford pet top-1 accuracies averaged across all architectures, columns are analogous to the Table 1 shown above for ImageNet.

	Upright training			Rotated training		AMR 300		AMR 50	
Testing	up	rot	% ceil	rot	% ceil	rot	% ceil	rot	% ceil
Stanford cars	0.867	0.165	19	0.618	71	**0.796**	**92**	0.746	86
Oxford pet	0.741	0.483	65	0.603	81	**0.712**	**96**	0.670	90

The best performances are bold.

rotated training. AMR-5 is slightly worse with 97% ceil, but it shows that it is possible to obtain an AMR module that is very useful with minimal training resources. Figure 3 contains polar plots that show the ImageNet top-1 accuracies of the different architecture families by angle. The solid orange lines show the accuracies of the upright-trained base models. We observe that the accuracies have their highest points at zero degrees rotation and then symmetrically drop off with increasing angle, reaching their lowest points at 135 and 225 degrees. We further observe that rotated training (green dash-dotted line) and AMR (blue dotted line) both achieve rotational invariance and exhibit performances that are independent of test time angles. Corresponding to the reported results in Table 1, AMR performance is consistently better than rotated training.

4.4 Stanford Cars and Oxford Pet

Due to Stanford Cars and Oxford Pet being smaller datasets we forgo ViTs and train the CNN models for more epochs on Stanford Cars and Oxford Pet. On Stanford Cars we train the base models for 10,00 epochs, and the corresponding AMR modules are trained for 300 and 50 epochs, respectively. On Oxford Pet, we train the base models for 3,000 epochs and the AMRs for 1,000 and 150 epochs. Table 2 shows the top-1 accuracies averaged across architectures and averaged across angles where appropriate (for full table see Table 7 in the Appendix) and Figure 4 polar plots, analogous to the ones for ImageNet in the above paragraph. Our core findings are replicated on both datasets: Rotating the images reduces all the models' performances and AMR remains the more powerful way of addressing rotations on these datasets. On Stanford Cars, the performance loss caused by rotations on the upright trained model is much more severe (19% ceil) compared to ImageNet (69% ceil), with the models failing almost completely when facing rotations larger than 20 degrees (see left column of Figure 4). This makes sense intuitively since cars are almost always upright in pictures with minimal variation, thus the models experience almost no variation during training. This is further supported by the observation that Oxford Pet, which is also a small dataset but contains animals that are naturally less static compared to cars, exhibits a milder drop off (65% ceil). We further observe that on Stanford Cars and to a lower extent on Oxford Pet, EfficientNets perform much better than ResNe(X)ts on rotated data, both with rotated training and AMR, while all architectures perform roughly equally well on upright data. We conjecture this is because EfficientNets have been designed to be sample efficient. This could allow them to train filters useful for a wide variety of tasks (such as AMR) even on a small dataset and a relatively short training time. However, an unexpected result is that AMR paired with ResNe(x)t models showed a decline in performance when approaching 0 degrees, while EfficientNets do not suffer from this effect. In the Appendix, we investigate this phenomenon further.

4.5 Comparison with existing rotation equivariant methods on MNIST

The focus of AMR is modern, large architecture and correspondingly large datasets. The current literature for rotation equivariant methods is focused on rotated MNIST as the

benchmark dataset of choice for most of these methods. To put our work into perspective with these related works we present the performances of ResNet-18 (He et al., 2016), ResNet-18 + rotated training and ResNet18+AMR on MNIST (see Table 3). The ceiling performance of ResNet18 on upright MNIST is almost one, which is to be expected. Similar to the larger datasets above is the performance of AMR superior to rotated training, however, only by a small margin. This makes sense intuitively since for such a simple dataset the reduced sample efficiency of rotated training plays a small role. Most importantly, the performance of ResNet18+AMR is competitive to the performances of the related works, which bake rotation invariance directly into their neural network designs.

4.6 Increase of computational cost incurred by AMR at inference time

The simple addon design of our AMR module enables the reuse of pre-trained base models, thereby saving a significant amount of training time. The AMR module itself is also very lightweight with 4 million parameter and 0.6 GFLOPS. One of the main drawbacks of our AMR method is that it requires two forward passes at inference time (one for the angle classification and one for the image classification). This leads to a total inference cost of two times the base network plus a fixed overhead for the AMR module. Table 4 shows the required parameter, flops and inference speed of the ResNet models with and without AMR. Unsurprisingly, the performance with AMR is drastically lower than without it. However, ResNet18+AMR is still much faster than ResNet50 while drastically outperforming it on rotated data.

TABLE 3 Top-1 accuracies on rotated MNIST for ResNet-18 based methods as well as related works, accompanied by ResNet-18 upright top-1 accuracy as a baseline.

Method	Top-1 Acc.
ResNet-18 (upright - ceil performance)	0.996
ResNet-18	0.48
ResNet-18 + rotated training	0.978
ResNet18 + AMR	0.981
Harmonic networks (Worrall et al., 2017)	0.983
Ti-pooling (Laptev et al., 2016)	0.988
G-CNNs (P4CNN) (Cohen and Welling, 2016)	0.972
RotEqNet (base) (Marcos et al., 2017)	0.989

4.7 AMR usefulness given the prevalence of rotated data

In an applied scenario, it is not always realistic that all inputs are presented at a random angle. We therefore investigate the usefulness of AMR when the test data consists of a combination of upright (up) and rotated (rot) images. To this end, we compute top-1 test errors on ImageNet of the ResNet family models on rotated and upright inputs separately. We repeat this process for upright training, rotated training and AMR-33 (see Table 5). We then linearly combine up and rot performances to obtain the final performances for mixed datasets consisting of both upright and rotated data. We increase the percentage of rotated data in the test mix until alternative methods (rotated training, AMR-33) start outperforming the default of upright training. We call percentages of parity between methods breakpoints (BP). Unsurprisingly, the BPs for rotated training (30% on average) are much higher than the ones of AMR-33 (7.5%). The key finding here is that BPs for AMR-33 are all below 10% which shows that only a small portion of the test set needs to be non-upright for AMR to be a worthwhile choice.

5 Validity of self-supervised training built on artificial rotation

Self-supervised learning based on artificial data modifications always warrants great caution. It is often unclear if the model learns to solve the desired task or if it simply learns to find unintended shortcuts in the self-supervision procedure. In our case, we use a digital rotation algorithm on our input images. While none are visible to the human eye, algorithm-specific artifacts are introduced to the rotated images. This raises the question if the AMR module

TABLE 5 Top-1 accuracies of ResNets on upright (up) and rotated (rot) ImageNet, accompanied with breakpoints (BP) that signify the share of rotated data in the test set necessary for alternative methods (rotated training, AMR) to outperform upright training.

	Upright Tr.		Rotated Tr.			AMR-33		
	up	rot	up	rot	BP	up	rot	BP
RN-18	69.5	43.3	59.3	58.9	35.5%	67.1	67.6	9.0%
RN-50	76.5	53.7	69.2	67.3	35.0%	75.1	75.5	6.1%
RN-152	77.9	55.2	73.6	73.0	19.5%	76.2	76.7	7.4%
Avg.	74.6	50.7	67.4	66.4	30%	72.8	73.3	7.5%

TABLE 4 The number of parameters (in millions), gigaflops at inference time as as well as inference speed for the ResNet family.

	RN18	RN50	RN152	RN18-AMR	RN50-AMR	RN152-AMR
No. parameter M	12	26	60	16	30	64
Inference flops GF	1.8	4	12	4.2	8.6	24.6
Inference speed image/sec	1970	474	319	700	233	168
Top-1 Acc. rot-ImageNet	0.433	0.537	0.552	0.676	0.755	0.767

Shown with and without AMR. For context we also repeat the top-1 accuracies of these models on rotated ImageNet.

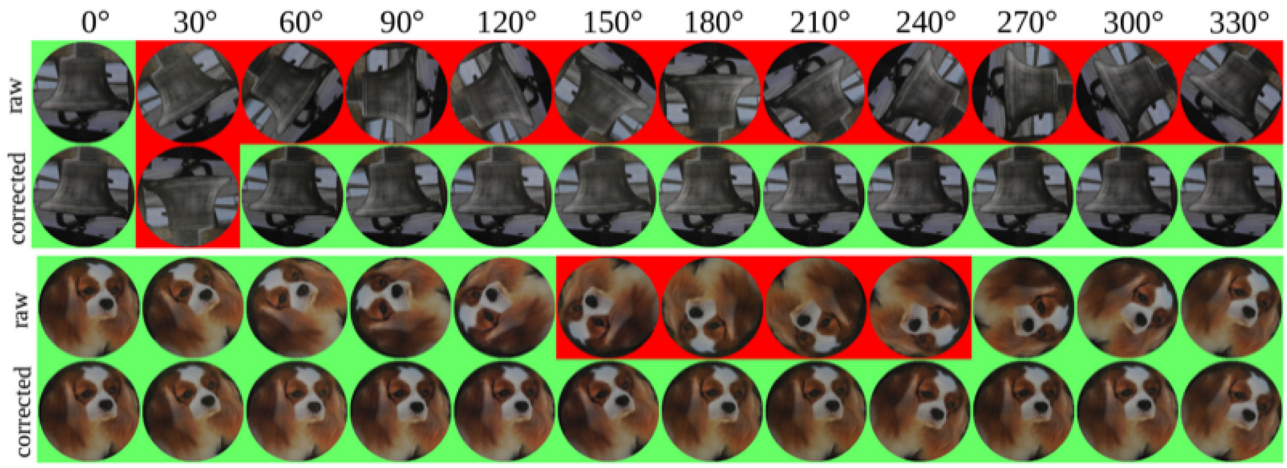

FIGURE 5
Photographs of two printed ImageNet validation samples taken at 12 different angles. Both samples are shown in their original state (raw) and after the mental rotation step (corrected). The color of the masked-out region indicates if the corresponding image has been correctly classified. The mental rotation and classification steps have been performed by ResNet50 + AMR33. Image taken from the ILSVRC2012 version of ImgeNet. https://www.image-net.org/challenges/LSVRC/2012/.

learns to classify the correct rotation angle based on unwanted traces of the rotation algorithm rather than by understanding the contents of the image. To ensure this is not the case, we perform the following ablation study: We print out seven images sourced from different classes from the ImageNet validation set. We then take photos of each of those printouts at twelve different in-plane rotations by physically rotating the print in 30-degree intervals. This way we guarantee the absence of any rotation algorithm artifacts that the model could have learned to use. The images were printed using a Konica Minolta bizhub 450i on maximum resolution with guidelines to enable accurate angular distances (see Figure 8 in the Appendix). The photos were then taken by hand using a Nikon Coolpix P7000 digital camera. Figure 5 shows all twelve re-digitized photos for two cases (raw). The color-coded background indicates if that photo was correctly classified by a standard trained ResNet50 base model (green denotes correct, red an error). We observe a similar effect as with Stanford Cars: Like a car, bells have a very clearly defined upright position. The bell, therefore, is only classified correctly when it is upright. Dogs on the other hand are very variable in appearance (e.g., head turned, laying down etc), thus the dog is only misclassified when it is completely upside down at rotations between 150 and 240 degrees. The second rows (corrected) show the outcome of applying Resnet50 + AMR33 to the above photos. The AMR module is able to correct the orientation of all but one photo. We conclude that it learned to classify the angles by understanding the image contents. While we cannot exclude any learning of artifacts introduced by the self-supervision process, if such are present, they do not hinder the training process from learning transferrable features. We further observe that the rotation correction is much more precise in the bell case than for the dog. This ties in with our assumption that the network's filters are much more precisely tuned to a sharp upright position for the bell compared to the dog. Across all 84 photos, the standard ResNet50 achieves a top-1 classification accuracy of 0.57. ResNet50 + AMR33, on the other hand, achieves a top-1 accuracy of 0.96, showing that the AMR module works properly on all printed images.

6 Application to a novel downstream task: semantic segmentation

Since they share the same neural network building blocks, the assumption that models for other vision tasks like object detection or semantic segmentation also struggle with rotated inputs suggests itself. In this section, we test this hypothesis and demonstrate how a trained AMR module can be used to easily improve the rotational stability of models for tasks other than classification. We choose semantic segmentation as an example. As the base model, we use a fully convolutional ResNet50 and source the matching pre-trained weights named "FCN_ResNet50_Weights.COCO_WITH_VOC_LABELS_V1" from torchvision. They have been trained on MSCoCo-Stuff (Lin et al., 2014), with a reduced class set only containing classes that are also available in PascalVOC (Everingham et al., 2015). We again mask the corners of all images. On this masked but upright data, the pre-trained model achieves a mean intersection over union (IoU) of 57.6. Rotating the images causes the mean IoU to drop to 32.7 (Table 6). This confirms our initial conjecture. We now take our ResNet50 + AMR33 which has been trained on ImageNet and use it to perform AMR steps (1) and (2) on CoCo without any additional retraining or modification. The angle-corrected inputs are then fed back into the base semantic segmentation model. This approach yields an IoU of 55.2, showing that AMR also works for semantic segmentation and that a trained AMR module can be easily transferred between similar datasets. Figure 6 shows two examples from the CoCo validation set with their corresponding ground truth, segmentation output, and AMR-corrected segmentation output, visually confirming our findings.

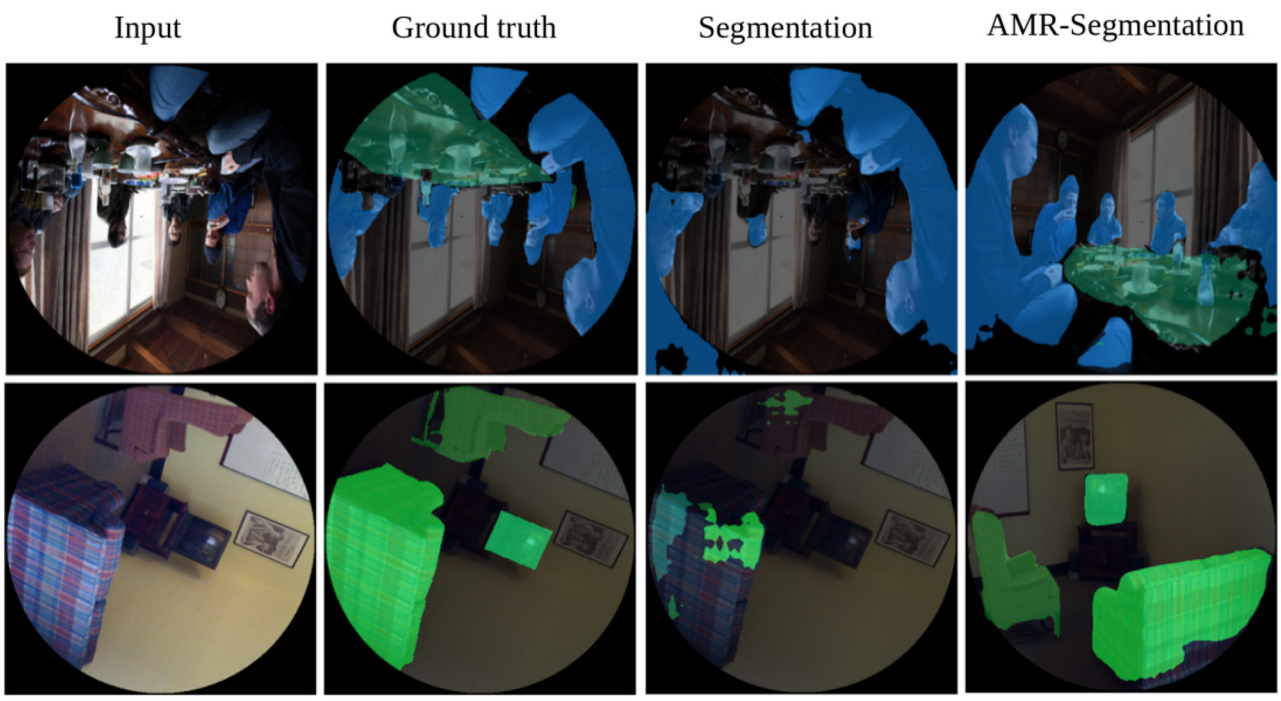

FIGURE 6
Two rotated examples (rows) from the CoCo validation set with their corresponding ground truth, segmentation output, and AMR-corrected segmentation output (columns). The segmentation on the rotated image exhibits bad performance in both samples. The column AMR-segmentation shows the output for the images that have been (correctly) un-rotated using AMR.

TABLE 6 Performances (in IoU) of FCN-ResNet50 on MSCoCo-Stuff on upright and rotated data as well as the performance of FCN-ResNet50+AMR on rotated data.

	FCN-ResNet50 Upright	FCN-ResNet50 Rotated	FCN-ResNet50+ AMR33 Rotated
Performance (IoU)	57.6	32.7	55.2

7 AMR as a canonicalization function

AMR can be seen as an implementation of a canonicalization function as described by Kaba et al. (2023). A core difference to existing implementations (Mondal et al., 2023) is that our canonicalization function is not learned but hard coded (in the form of the rotation algorithm present in open-CV). It is then individually parameterized per input in a one shot fashion by a learned function—the AMR module. Our architecture encodes a strong inductive bias for in-plane rotations only. This causes the AMR module to be easily trainable within a few epochs, representing a large advantage over training a model to predict canonicalization functions directly, which is empirically difficult to optimize in practice as described by Mondal et al. (2023). However, employing such a strict inductive bias is a trade-off since it will most likely lead to sub-optimal learning in the presence of transformations that can not be encoded within that bias (e.g., rotations off-plane from the image itself). Most datasets such as ImageNet likely contain a wide variety of such perturbations. They can make the self-supervised learning signal more noisy but our results show that they do not critically impair the AMR training.

8 Limitations and future work

A key drawback of AMR is that two forward passes are necessary for inference. This is part of the core design and cannot be changed. It is mitigated partially by the fact that a smaller model can be chosen in conjunction with AMR and still outperform a large model trained with rotational data augmentation due to the inefficiency of that approach resulting in a less costly package even at test time. For example, a forward pass through the AMR module and 2xResNet-50 is 8.6 GFlops, whereas a single forward pass through a ResNet-152 is 12 GFLOPS (see Table 4); still, the AMR-combination outperforms the larger ResNet in this example (see Table 1). With applicability in mind, we opted to focus on 2D in-plane rotations of whole images featuring one dominant object. Our work is, therefore, not suited for cases where multiple objects are individually rotated. This scenario could be addressed by combining a region-proposal-based method such as Faster-RCNN (Girshick, 2015) with AMR at the proposal level. In the real world, 3D objects are rotated in 3D space, which can lead to much more drastic changes in appearance. Extending AMR to this realistic setting (i.e., at the hands of a game engine or interactive learning using robots in the real world) would be a very promising, most natural extension of this work that could lead to vision systems that learn more complete and abstract representations of objects. An exciting future application for AMR models would be reducing the rotational variability of an existing

dataset (e.g., ImageNet, by making all appearing objects upright). This would further disentangle the training of upright appearances from rotations which would likely lead to improved training efficiency of base models.

9 Conclusions

We have presented AMR, a neuropsychology-inspired approach for handling rotated data in vision systems. We have shown that AMR consistently outperforms the most common technique of rotational data augmentation across different deep architectures and datasets. We have shown the viability of AMR in realistic cases where the data is a mixture of upright and rotated inputs. We further presented a sanity check which confirms that our self-supervised learning setup learns to identify rotations by the content of the images. Lastly, we have shown how a trained AMR module can easily be transferred to another model built for a different task to improve its rotational stability, underpinning its flexibility to be used with any architecture.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

LT: Visualization, Data curation, Validation, Project administration, Writing – original draft, Formal analysis, Methodology, Software, Conceptualization, Investigation, Resources, Writing – review & editing. TS: Project administration, Writing – review & editing, Writing – original draft, Funding acquisition, Supervision. JS: Supervision, Writing – original draft, Funding acquisition, Conceptualization, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work has been financially supported by grants 34301.1 IP-ICT "RealScore" (Innosuisse) and ERC Advanced Grant AlgoRNN nr. 742870. Open access funding by Zurich University of Applied Sciences (ZHAW).

Acknowledgments

Thanks go to Mohammadreza Amirian and Philipp Denzel for insightful discussions and Jasmina Bogojeska for the idea of the sanity check.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2025.1644044/full#supplementary-material

References

Allingham, J. U., Mlodozeniec, B. K., Padhy, S., Antorán, J., Krueger, D., Turner, R. E., et al. (2024). A generative model of symmetry transformations. *arXiv preprint arXiv:2403.01946*. doi: 10.48550/arXiv.2403.01946

Baird, H. S. (1992). "Document image defect models," in *Structured Document Image Analysis* (Berlin: Springer), 546–556. doi: 10.1007/978-3-642-77281-8_26

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*. doi: 10.48550/arXiv.2108.07258

Boominathan, L., Srinivas, S., and Babu, R. V. (2016). "Compensating for large in-plane rotations in natural images," in *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing* (New York, NY: ACM), 1–8. doi: 10.1145/3009977.3009988

Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal of Software Tools*. 120, 122–125.

Cohen, T., and Welling, M. (2016). "Group equivariant convolutional networks," in *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2990–2999. Available online at: JMLR.org

Cohen, T. S., and Welling, M. (2022). "Steerable CNN,". in *International Conference on Learning Representations*. Available online at: OpenReview.net

Dieleman, S., Willett, K. W., and Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Mon. Not. R. Astron. Soc.* 450, 1441–1459. doi: 10.1093/mnras/stv632

Ding, W., and Taylor, G. W. (2014). "Mental rotation" by optimizing transforming distance. *arXiv preprint arXiv:1406.3010*. doi: 10.48550/arXiv.1406.3010

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). "An image is worth 16 × 16 words: transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR*. Available online at: OpenReview.net

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. (2019). "Exploring the landscape of spatial robustness," in *Proceedings of the 36th International Conference on Machine Learning, ICML*, 1802–1811. Available online at: JMLR.org

Esteves, C., Allen-Blanchette, C., Zhou, X., and Daniilidis, K. (2018). "Polar transformer networks," in *International Conference on Learning Representations*. Available online at: OpenReview.net

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2015). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. Available online at: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (Accessed August 10, 2024).

Fang, J., Zhou, D., Song, X., Jin, S., Yang, R., and Zhang, L. (2020). "Rotpredictor: unsupervised canonical viewpoint learning for point cloud classification," in *2020 International Conference on 3D Vision (3DV)* (IEEE), 987–996. doi: 10.1109/3DV50981.2020.00109

Feng, Z., Xu, C., and Tao, D. (2019). "Self-supervised representation learning by rotation feature decoupling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Computer Vision Foundation/IEEE), 10364–10374. doi: 10.1109/CVPR.2019.01061

Fukushima, K., Miyake, S., and Ito, T. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern*. 36, 193–202. doi: 10.1007/BF00344251

Fukushima, K., Miyake, S., and Ito, T. (1983). Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Trans. Syst. Man Cybern*. 13, 826–834. doi: 10.1109/TSMC.1983.6313076

Girshick, R. (2015). "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE Computer Society), 1440–1448. doi: 10.1109/ICCV.2015.169

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (Computer Vision Foundation/IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Immer, A., van der Ouderaa, T., Rätsch, G., Fortuin, V., and van der Wilk, M. (2022). Invariance learning in deep neural networks with differentiable laplace approximations. *Adv. Neural Inf. Process. Syst*. 35, 12449–12463.

Jaderberg, M., Simonyan, K., Zisserman, A., and kavukcuoglu, K. (2015). "Spatial transformer networks," in *Advances in neural information processing systems 28*. Available online at: https://proceedings.neurips.cc//

Kaba, S.-O., Mondal, A. K., Zhang, Y., Bengio, Y., and Ravanbakhsh, S. (2023). "Equivariance with learned canonicalization functions," in *International Conference on Machine Learning* (Honolulu: PMLR), 15546–15566.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al. (2023). "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE Computer Society), 4015–4026. doi: 10.1109/ICCV51070.2023.00371

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). "3D object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)* (IEEE Computer Society), 554–561. doi: 10.1109/ICCVW.2013.77

Laptev, D., Savinov, N., Buhmann, J. M., and Pollefeys, M. (2016). "Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (IEEE Computer Society), 289–297. doi: 10.1109/CVPR.2016.38

Leclerc, G., Ilyas, A., Engstrom, L., Park, S. M., Salman, H., and Madry, A. (2022). *FFCV: Accelerating Training by Removing Data Bottlenecks*. Available online at: https://github.com/libffcv/ffcv/ (Accessed August 10, 2024).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: common objects in context," in *Computer Vision-ECCV 2014: 13th European Conference* (Berlin: Springer), 740–755. doi: 10.1007/978-3-319-10602-1_48

Marcos, D., Volpi, M., Komodakis, N., and Tuia, D. (2017). "Rotation equivariant vector field networks," in *IEEE International Conference on Computer Vision, ICCV* (IEEE Computer Society), 5058–5067. doi: 10.1109/ICCV.2017.540

Miao, N., Rainforth, T., Mathieu, E., Dubois, Y., Teh, Y. W., Foster, A., et al. (2023). "Learning instance-specific augmentations by capturing local invariances," in *International Conference on Machine Learning* (Honolulu: PMLR), 24720–24736.

Mondal, A. K., Panigrahi, S. S., Kaba, O., Mudumba, S. R., and Ravanbakhsh, S. (2023). Equivariant adaptation of large pretrained models. *Adv. Neural Inf. Process. Syst*. 36, 50293–50309.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. (2012). "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (Computer Vision Foundation/IEEE). doi: 10.1109/CVPR.2012.6248092

Quiroga, F., Ronchetti, F., Lanzarini, L., and Bariviera, A. F. (2020). "Revisiting data augmentation for rotational invariance in convolutional neural networks," in *Modelling and Simulation in Management Sciences: Proceedings of the International Conference on Modelling and Simulation in Management Sciences (MS-18)* (Berlin: Springer), 127–141. doi: 10.1007/978-3-030-15413-4_10

Rojas-Gomez, R. A., Lim, T.-Y., Do, M. N., and Yeh, R. A. (2024). "Making vision transformers truly shift-equivariant," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Computer Vision Foundation/IEEE), 5568–5577. doi: 10.1109/CVPR52733.2024.00532

Romero, D. W., and Cordonnier, J. (2021). "Group equivariant stand-alone self-attention for vision," in *9th International Conference on Learning Representations, ICLR*. Available online at: OpenReview.net

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis*. 115, 211–252. doi: 10.1007/s11263-015-0816-y

Schlag, I., Irie, K., and Schmidhuber, J. (2021). "Linear transformers are secretly fast weight programmers," in *International Conference on Machine Learning, ICML*, 9355–9366. Available online at: JMLR.org

Schmidhuber, J. (1992). Learning to control fast-weight memories: an alternative to dynamic recurrent networks. *Neural Comput*. 4, 131–139. doi: 10.1162/neco.1992.4.1.131

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., et al. (2019). Adversarial training for free! *Adv. Neural Inf. Process. Syst*. 32, 3353–3364.

Shepard, R. N., and Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science* 171, 701–703. doi: 10.1126/science.171.3972.701

Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). "Best practices for convolutional neural networks applied to visual document analysis," in *7th International Conference on Document Analysis and Recognition (ICDAR)* (IEEE Computer Society), 958–962. doi: 10.1109/ICDAR.2003.1227801

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *arXiv preprint arXiv:1505.00387*. doi: 10.48550/arXiv.1505.00387

Stadelmann, T., Amirian, M., Arabaci, I., Arnold, M., Duivesteijn, G. F., Elezi, I., et al. (2018). "Deep learning in the wild," in *Artificial Neural Networks in Pattern Recognition: 8th IAPR TC3 Workshop* (Berlin: Springer), 17–38. doi: 10.1007/978-3-319-99978-4_2

Stadelmann, T., Tolkachev, V., Sick, B., Stampfli, J., and Dürr, O. (2019). "Beyond imagenet: deep learning in industrial practice," in *Applied Data Science: Lessons Learned for the Data-Driven Business* (Springer: New York), 205–232. doi: 10.1007/978-3-030-11821-1_12

Tai, K. S., Bailis, P., and Valiant, G. (2019). "Equivariant transformer networks," in *International Conference on Machine Learning* (PMLR), 6086–6095.

Tan, M., and Le, Q. (2019). "Efficientnet: rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning, ICML*, 6105–6114. Available online at: JMLR.org

TorchVision Maintainers and Contributors. (2016). *Torchvision: Pytorch's Computer Vision Library*. Available online at: https://github.com/pytorch/vision (Accessed August 10, 2024).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst*. 30, 5998–6008.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech Signal Process*. 37, 328–339. doi: 10.1109/29.21701

Weiler, M., and Cesa, G. (2019). "General e (2)-equivariant steerable CNNS," in *Advances in Neural Information Processing Systems 32*. Available online at: https://proceedings.neurips.cc//

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). "Harmonic networks: Deep translation and rotation equivariance," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (Computer Vision Foundation/IEEE), 5028–5037. doi: 10.1109/CVPR.2017.758

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR* (Computer Vision Foundation/IEEE), 1492–1500. doi: 10.1109/CVPR.2017.634

Xu, R., Yang, K., Liu, K., and He, F. (2023). "$e(2)$-equivariant vision transformer," in *Uncertainty in Artificial Intelligence* (Pittsburgh, PA: PMLR), 2356–2366.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). "Scaling vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Computer Vision Foundation/IEEE), 12104–12113. doi: 10.1109/CVPR52688.2022.01179

Zhang, J., Li, G., Su, Q., Cao, L., Tian, Y., and Xu, B. (2025). Enabling scale and rotation invariance in convolutional neural networks with retina like transformation. *Neural Netw*. 187:107395. doi: 10.1016/j.neunet.2025.107395

Zhang, W., Tanida, J., Itoh, K., and Ichioka, Y. (1988). Shift-invariant pattern recognition neural network and its optical architecture. *Proc. Annu. Conf. Jpn. Soc. Appl. Phys*. 564, 6p-M-14.