SN Computer Science (2021) 2:340
https://doi.org/10.1007/s42979-021-00735-0

SURVEY ARTICLE

CNN Architectures for Geometric Transformation-Invariant Feature Representation in Computer Vision: A Review

Alhassan Mumuni[1] · **Fuseini Mumuni**[2]

Received: 22 January 2021 / Accepted: 7 June 2021 / Published online: 16 June 2021

Abstract
One of the main challenges in machine vision relates to the problem of obtaining robust representation of visual features that remain unaffected by geometric transformations. This challenge arises naturally in many practical machine vision tasks. For example, in mobile robot applications like simultaneous localization and mapping (SLAM) and visual tracking, object shapes change depending on their orientation in the 3D world, camera proximity, viewpoint, or perspective. In addition, natural phenomena such as occlusion, deformation, and clutter can cause geometric appearance changes of the underlying objects, leading to geometric transformations of the resulting images. Recently, deep learning techniques have proven very successful in visual recognition tasks but they typically perform poorly with small data or when deployed in environments that deviate from training conditions. While convolutional neural networks (CNNs) have inherent representation power that provides a high degree of invariance to geometric image transformations, they are unable to satisfactorily handle nontrivial transformations. In view of this limitation, several techniques have been devised to extend CNNs to handle these situations. This article reviews some of the most promising approaches to extend CNN architectures to handle nontrivial geometric transformations. Key strengths and weaknesses, as well as the application domains of the various approaches are also highlighted. The review shows that although an adequate model for generalized geometric transformations has not yet been formulated, several techniques exist for solving specific problems. Using these methods, it is possible to develop task-oriented solutions to deal with nontrivial transformations.

Keywords Convolutional neural network · Robust computer vision · Invariant recognition · Transformation-equivariant network · Symmetry group transformation

Introduction

Background

Geometric transformation invariance is the ability of feature representation in a computer vision model to remain unchanged under geometric transformations of input images that result from visual appearance changes of the underlying objects. The significance of geometric transformation invariance stems from the fact that the real world is inherently three-dimensional (3D), and object appearances can change drastically depending on their orientation in the 3D world, proximity, camera viewpoint or perspective. In addition, phenomena such as occlusion, deformation and foreground clutter can result in significant geometric appearance changes of objects. This ultimately leads to various geometric transformations of the resulting images that affect the performance of computer vision models (see Fig. 1). Under these conditions, it is desirable that the underlying feature representations remain invariant. In classification and object tracking applications, for example, the need for geometric transformation invariance is obvious, since the machine vision system is generally required to maintain a consistent interpretation of objects at different scales, orientations and shapes resulting from factors such as changing camera view angles, occlusions, deformations and spatial orientations. In a real-world setting the same object can be located at different positions in an image (as in Fig. 2), resulting in different pixel representations of the semantically identical images.

✉ Alhassan Mumuni
alhassan.mumuni@cctu.edu.gh

[1] Electrical & Electronic Engineering Department, Cape Coast Technical University, Cape Coast, Ghana

[2] Electrical and Electronic Engineering Department, University of Mines and Technology, Tarkwa, Ghana

Fig. 1 Examples of geometric transformations that make seemingly easy recognition tasks challenging for computer vision algorithms. The above is a demonstration by Alcorn et al. [1] of prediction performance of Inception—V3 under different viewing conditions of a **a** fire truck and **b** school bus. Images courtesy Ref. [1], © IEEE 2019

In this case, for example, the fox in the images is the same irrespective of whether it is in the middle or at one corner of the image. Similarly, a zebra upside down is still a zebra and a machine vision model should still be able to recognize it as such. Unfortunately, these problems are currently challenging tasks for visual recognition approaches based on state-of-the-art deep learning models [1].

Another useful property of visual recognition systems is equivariance—the ability to change the representation of learned objects in a way that corresponds to the observed transformations of the underlying objects [2]. In convolutional neural networks, this means performing feature extraction operations (e.g., convolution, activation and pooling) over a transformed image results in a corresponding transformation of the generated output feature vectors. The equivariance of the neural network with respect to translation, rotation and scale is depicted in Fig. 2.

In contrast to invariant representation (Fig. 2c), where the goal is to ignore the effects of transformations, equivariant representation (Fig. 2d) reflects all visual changes associated with transformations of the input image. In Fig. 2d, for example, to ensure rotation equivariance, it is necessary to replicate rotation of the input image at the output. Equivariance of convolutional neural networks to geometric transformations ensures that no geometric information such as object size (scale), pose, position or deformation is lost in the deep learning pipeline. This is important in visual recognition tasks like scene understanding, semantic visual SLAM, robot navigation, object manipulation and pose estimation.

Overview of Approaches to Geometric Transformation Invariance

The earliest approaches to geometric transformation-invariant recognition relied on part-based representations [3]

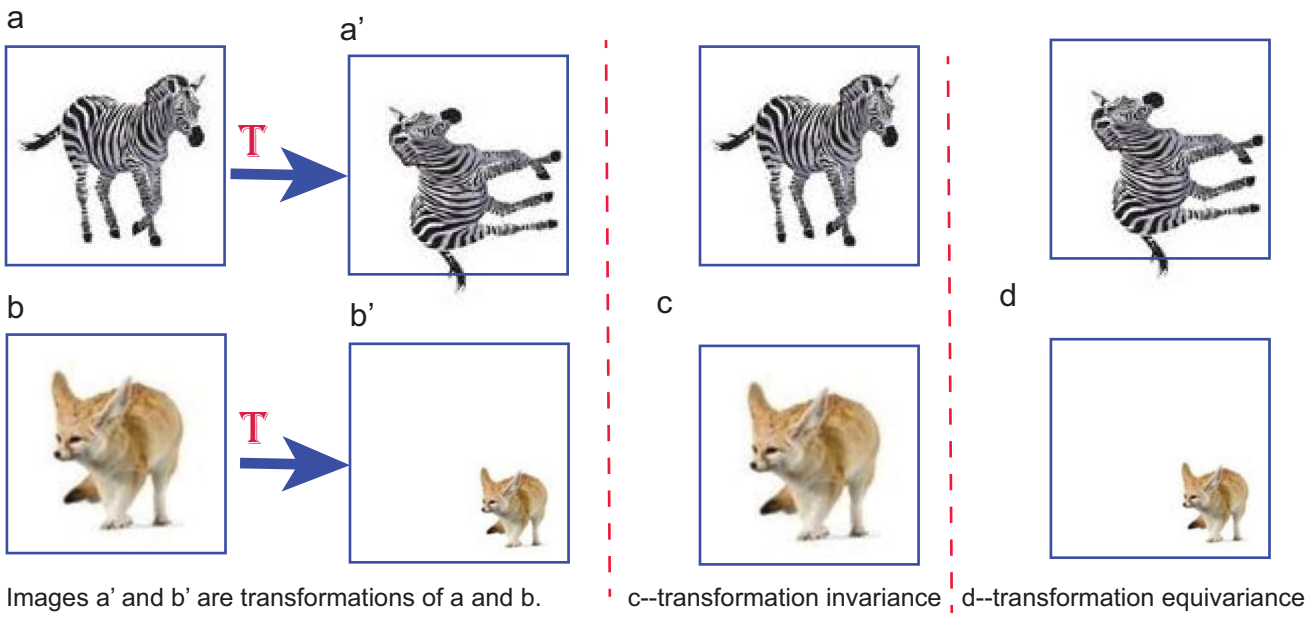

Fig. 2 Invariance vs equivariance: the original image has been transformed by a 90 degree rotation (**a'**) and a 1/3 scaling plus translation (**b'**). The two panels on the right depict transformation-invariant (**c**) and -equivariant (**d**) representations. The goal of an invariant model is to interpret each as a canonical image regardless of the transformation. On the other hand, an equivariant representation must show the corresponding changes in the output

and explicit geometric models [4]. Later, machine learning approaches soon replaced hard-coded, analytical models as they provided a more general approach to dealing with a wide variety of tasks. Traditional machine learning approaches to tackling invariant visual recognition problems such as image classification, object detection and scene segmentation involve the use of hand-crafted feature descriptors—for example, SURF, RANSAC, HOG, SIFT, LBP or ORB—in conjunction with one or more machine learning techniques such as Support Vector Machines (SVM), Conditional Random Fields (CRFs), Hidden Markov Models (HMMs), Decision Trees, Random Forests and Principal Component Analysis (PCA) [5].

In recent years, however, deep learning approaches have become the most dominant means for solving these challenging machine vision problems, achieving much higher accuracies than traditional machine learning methods in many recognition tasks [6, 7]. Deep learning approaches are characterized by training with very large annotated data using a single multi-layer neural network model as an optimization algorithm. Low-level features are learned automatically from input data without the need for hand-crafted feature engineering. Presently, convolutional neural networks (CNNs) outperform other deep learning methods in most machine vision applications [8].

The basic architecture of the CNN was originally proposed in 1988 by Fukushima [9] but its application was limited by the complexity of existing training approaches. It was not until 1998, following the successful application of gradient descent for training CNN (LeNet) by Yan LeCun [10] that CNNs started seeing widespread adoption for solving practical machine vision problems. Interest in CNNs again surged with the groundbreaking results of AlexNet by Krizhevsky et al. [7]. The approach made use of a combination of previously developed techniques such as dropout [11], stochastic gradient descent [12], rectified linear (ReLu) activation function [13], spatial pooling [14], and weight decay and momentum [15]. In addition, the use of GPUs for parallel processing allowed the practical training on very large ImageNet data set. A CNN architecture, to a large degree, corresponds to the basic functional structure of biological visual systems [16–19] and retains a number of its important properties. In particular, it employs spatially shared weights to learn invariant features. As a result, the CNN possesses high generalization ability in image domains and is able to handle geometric transformations [20], especially translation [18] and small changes in viewpoint [19]. In addition, the architecture is highly modular and can be readily used with other machine learning models. Compared with fully connected multilayer neural networks, CNNs are characterized by extremely fast learning and inference speed. These features make CNN architectures promising for further developments and exploitation as generic tools for solving geometric-transformation invariant recognition problems.

Basic Structure and Operation of Convolutional Neural Networks

Essentially, a CNN is made up of three main components: alternating arrangement of convolution and pooling layers, and a regular fully connected neural network. Convolution layers constitute the main functional elements of the CNN. The basic structure of CNN model is shown in Fig. 3b. Perceptrons in Convolution layers are divided into small blocks that learn a common weight during training. In addition to weight sharing, synaptic transmission of information is restricted by local spatial connectivity [24] similar to biological visual cortex [16–19], i.e., individual neurons are connected to local input areas, called receptive fields. These receptive fields overlap, covering the entire input image. This organization contrasts with conventional multilayer neural networks, which have no such spatial constrain, and each neuron has its own weight and connects to all other neurons of the immediate neighboring layer (Fig. 3a). The division of neurons into smaller local blocks allows different sets of neurons to specialize in learning specific image features.

Padding the borders of the input image with extra pixels is a common technique to prevent loss of information at locations close to the edges of the image. Padding also preserves the input dimension after applying convolution. The values of the extra (padded) pixels are usually set to zero so that they do not affect the result of the convolution operation. Another technique commonly used with convolution is striding [25]. The stride controls the number of steps (i.e., the number of elements that are skipped) between successive convolutions. A stride value of 1 is the default convolution and results in the maximum overlap of receptive fields. Bigger strides reduce redundant information in neighboring receptive fields and are usually employed to reduce computational cost but in some cases they may also help to generalize features better [26].

Convolutional layers use spatial filtering to learn meaningful image features for high-level recognition. This is accomplished by sliding a window, often called a filter or kernel—which is basically a square matrix of weights—over the entire input image to produce intermediate pixel blocks called feature maps. This process, known as convolution, transforms the spatial frequency characteristics of the input image. Subsequent layers following the first convolution layer extract features from already generated feature maps.

The result of each convolution operation is further processed by applying nonlinear activation. The activation is generally considered a se parate layer but it is logically part of the convolution layer. Neurons in each convolution layer

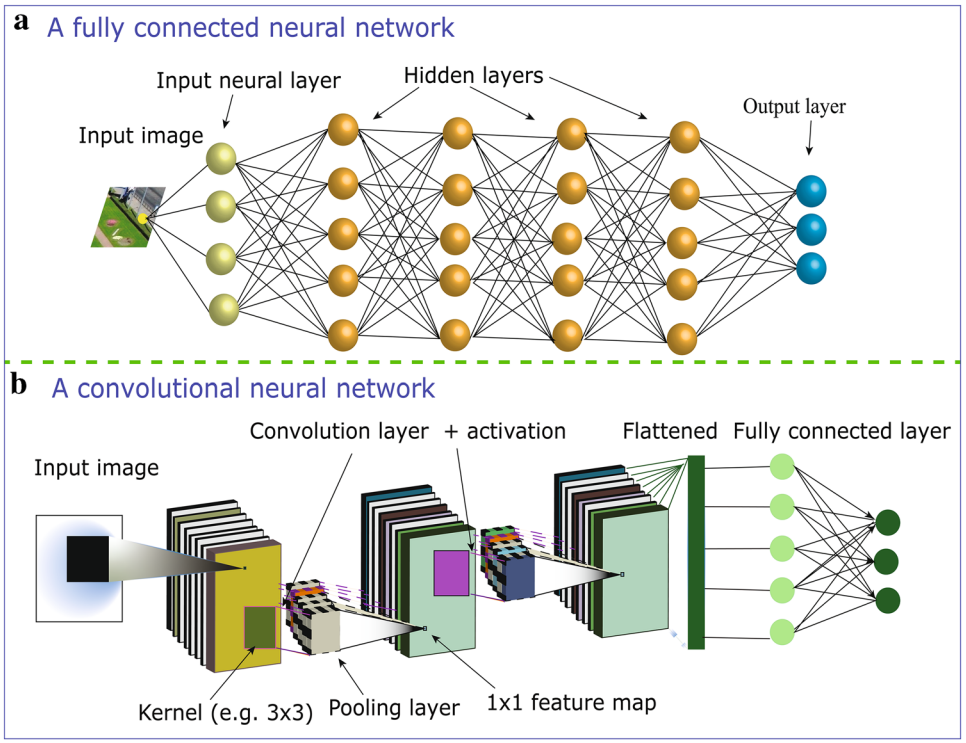

Fig. 3 Basic structure of a fully connected multi-layer network (**a**) and a convolutional neural network (**b**)

are connected to a subsampling layer, also called pooling layer, where so-called pooling is performed before feeding the feature maps to the next convolution layer. The most common pooling methods are max pooling [27] and average pooling [28]. Recently, many different pooling methods have been proposed to improve generalization in specific scenarios [29]. In addition to increasing generalization performance, the pooling operation reduces the spatial dimension of feature maps, leading to a significant acceleration of the learning process and inference.

After going through several layers of convolution and pooling, the learned features are combined and fed to a regular fully connected neural network, which may itself consist of several layers. The fully connected layers map high-level features to semantic labels. They essentially convert the reduced 2D feature vectors into scalar values (i.e., 1D feature vectors). The fully connected layers usually contain the bulk of the tunable parameters of a CNN. CNNs are trained using gradient descent methods [30], usually the error-based backpropagation algorithms.

Scope and Outline of Survey

The focus of this paper is on monocular vision techniques (i.e., approaches based only on 2D images as input data). Although deep learning models based on 3D or RGB-D images are becoming popular in visual recognition tasks—and in some cases they provide better recognition performance than 2D approaches, especially in tasks such as depth perception, shape analysis and scene reconstruction [21]—computational requirements and scarcity of training data limit their utility in applications such as robotics and visual SLAM. Moreover, the simplicity of 2D images make them more compelling for recognition tasks. This paper presents neural architectural design techniques that exploit spatial topology of 2D images and their correlation with CNN components such as 2D convolution filters and feature maps. A discussion of common data sets and performance comparisons are out of scope of the current work. In addition, deep learning approaches that utilize representation priors in the form of compositional part models [22] and realistic computer graphics models [23] are not covered in this survey. Instead, the main focus is on approaches that rely on internal representation schemes and special architectural configurations of CNNs for learning geometric transformations.

The remainder of this paper is set out as follows. In "Geometric Transformation Invariance in Deep Convolutional Neural Networks", we present a general overview of invariant feature representation in deep CNNs. A comprehensive discussion of state-of-the-art techniques for encoding geometric transformation in CNN models is presented in "Specialized CNN Architectures for Geometric Transform-Invariant Representation". The surveyed approaches have been grouped into three broad categories. The first group of approaches is primarily focused on architectures that embed special elements into CNN models to model affine and arbitrary geometric transformations. The second group of methods model single transformations: rotation, scale

and projective transformations. Lastly, the third family of approaches focuses mainly on techniques that exploit group theory to provide a compact, invariant representations for a specific set of geometric transformations. "Emerging Trends and Future Research Directions" briefly discusses current trends and future research directions. In "Summary and Conclusion", we conclude by summarizing the main issues covered in the paper.

Geometric Transformation Invariance in Deep Convolutional Neural Networks

Transformation-Invariant Feature Representation in CNN

The main mechanisms responsible for transformation invariance in CNN are the convolution and subsampling operations. As already mentioned above, each filter has shared weights which essentially allows it to learn its own instance of a feature map. In general, filters containing different weights generate different features under convolution. Since there are usually thousands of filters in the convolution layers, several independent feature maps are generated for each layer. The CNN is fundamentally endowed with translation invariance, because the feature maps generated by convolution are shifted over the entire pixel space—allowing useful features to be detected irrespective of their location in the image.

In the pooling layers, local averaging of neighboring pixels is performed. The pooling operation is essentially equivalent to "summarizing" the most important image features learned in the preceding layers [14]. Since pooling typically returns one representative value from a feature map (often the maximum of each feature vector), the result will practically be unchanged even when the position of this pixel changes, provided it is still within the receptive field under consideration. Thus, pooling provides additional generalization of various image transformations such as small changes in the position of image features in previous layers, and image distortions [29]. It increases the robustness of the output feature maps to minor deformations and small variations in image structure. Moreover, repeated convolution and pooling allows the network to progressively learn image features—from simple (low-level visual features such as lines, curves, corners, edges and basic textures) to more complex features like shapes, and finally to more abstract, high level concepts like whole objects such as faces and cars. Thus, higher up the CNN layers, the "visibility" of the receptive fields expand, allowing the network to capture high-level structure from the input image. The alternating process of convolution and pooling also provide some degree of scale-invariance as the kernel size is varied to capture features at different scales. In particular, the pooling operation expands the receptive field of the network (i.e., the size of the effective area in the input image that produces the feature maps) [31, 32] without correspondingly increasing computational load. This helps to ensure that sufficient receptive field sizes that encompass all relevant image regions are produced. Indeed, the expansion of the size of receptive field has been empirically shown [32] to improve invariant generalization performance of CNNs.

Common Approaches for Improving Generalization in CNNs

The main goal of a neural network model is to extract useful, generic relationships from training data that allow it to generalize well to new, unseen data. In the context of visual recognition, this also means the ability to generalize well to geometric as well as photometric transformations of previously trained images. Despite their higher performance over other deep learning methods in visual recognition tasks, conventional CNNs are not naturally invariant to image transformations that deviate significantly from training conditions [33]. A common approach to improve network generalization is to increase the network capacity by increasing its depth (that is, number of layers) and width (i.e., the number of nodes per layer). However, in practice, a particularly common problem that arises when training with large networks is overfitting, a situation, where the network accurately learns the input training data but shows a large error on unseen data. Many different techniques have been devised to tackle this problem (see Fig. 4).

One obvious way to overcome this limitation is to use smaller networks and incorporate additional context information in the form of representation priors such as deformable part models [22, 34–36] or realistic physics engines [23, 37, 38] that model causality. However, this approach is tedious and impractical for complex problems.

Several strategies have been proposed to improve generalization and overcome overfitting without explicitly incorporating hard-coded context information. Geometric transformation invariance can be accomplished by some of these general strategies that have been developed for invariant generalization in deep learning networks (Fig. 3). These include data augmentation, pre-training, transfer learning, meta-learning and few-shot learning techniques. In addition, regularization and parameter optimization strategies are essential for robust generalization. Special activation functions [39–41] and pooling methods [42–45], as well as regularization techniques [11, 46–48] are often used to improve generalization. Pooling methods, in particular, have been widely investigated as a means for achieving generalization with respect to geometric image transformations. Common pooling methods for this task include spatial pyramid

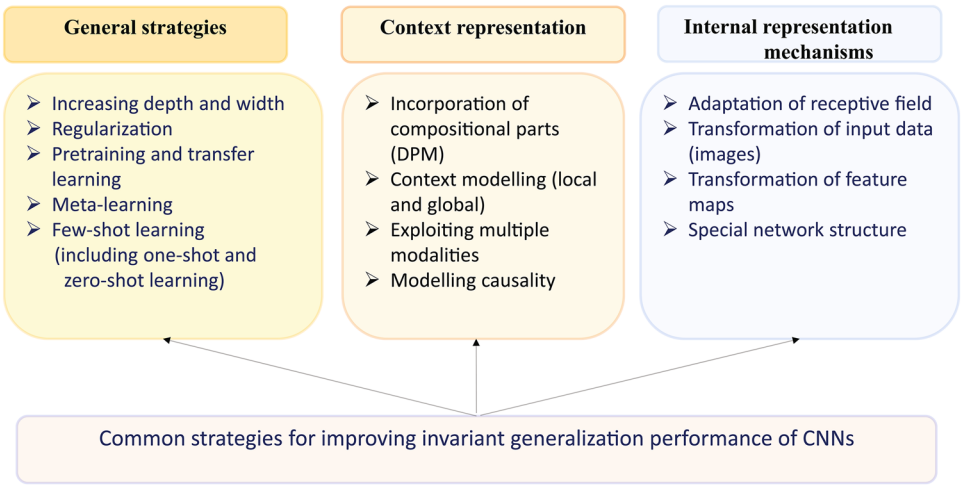

Fig. 4 Main strategies that have been proposed to solve invariant generalization problems in machine vision domains can be divided into three broad categories: (1) general approaches that reduce overfttting, (2) methods that exploit additional context to enrich the network model and (3) internal representation techniques that specifically exploit the properties of CNN elements and their correlation with image data

pooling (SPP) [42], transformation-invariant pooling [43], Polynomial Pooling [49], Lp pooling [50], def-pooling [51] and region-of-interest (RoI) pooling [52]. Because of the limitations of current approaches, data augmentation strategies [33, 53, 54]—where different transformed variations of the original images are produced to complement the training data—are usually employed in conjunction with the aforementioned techniques. CNN models using various combination of these techniques have led to rapid improvements in CNN models, driving accuracies from 83.6 percent. (AlexNet [7]) to a massive 96.5 percent (ResNet [6]) in large-scale image classification tasks (in just about 3 years). While these recent innovations in traing approaches have led to dramatic improvement of recognition accuracy in large-scale machine vision tasks, developing sufficiently robust CNN models to handle non-trivial transformations in challenging domains remains very problematic [55, 56]. Consequently, the search for techniques to handle geometric image transformations is still a very active research pursuit.

Recently, specialized CNN architectures that employ various internal representation techniques to explicitly model geometric transformations have been proposed. The approaches employ different strategies to achieve geometric transformation invariance: explicit transformation of input data and feature maps, special configurations of convolution layers, and flexible adaptation of receptive fields. Some of the most important approaches surveyed in this paper, and the main mechanisms underlying their functional principles of operation, are summarized in Table 1. In the following section (i.e., "Specialized CNN Architectures for Geometric Transform-Invariant Representation"), we provide a detailed review of the approaches that rely on internal representation of invariant features in convolutional neural networks. Key strengths and limitations, as well as common applications of the various approaches are also highlighted. We have categorized the surveyed methods into three broad classes. This categorization is based on the universality of the approaches in particular application domains.

Specialized CNN Architectures for Geometric Transform-Invariant Representation

In general, approaches to geometric-transformation invariance involve modifying the main functional elements (e.g., filters) in conventional CNNs or embedding special functional modules (e.g., analytical operators) or employing special network configurations (e.g., multi-nested synaptic

Table 1 Approaches to tackling transformations in general settings

Feature representation mechanism	Representative works	Application setting
Adaptive control of receptive fields	DCN [58], DFN [59], ACN [71]	Arbitrary transformations
Explicit transformation of receptive fields or feature maps	STN [89], IC-STN [110] Ref. [60] TICNN [104]	Affine transformations
Deeply learned transformations	Refs. [112, 113]	Arbitrary transformations
Parallel, multi-stream network organization	MC-STCNN [118] Refs. [114, 116, 119]	Arbitrary or affine transformations
Special network structure	CapsNet [57, 75]	Arbitrary transformations

connections) to explicitly handle geometric transformations. These techniques endow CNNs with the ability to generalize without the need for introducing transformed images into training data sets through data augmentation. In this section, we classify these approaches into three categories based on the universality of the methods (Fig. 5). The first group of approaches [57–60] attempt to tackle invariance in a holistic way; they model a broad range of generic transformations. The goal of these approaches is to enable CNN models to formulate a general concept of a broad range of geometric image transformations when presented with only one image of an object. The second group of approaches (e.g., [64–68]) are designed to primarily tackle single transformations. These approaches are generally simpler and more lightweight than the other two approaches discussed, and are, therefore, widely used in commonly available pre-trained scene segmentation, object detection and classification models. The third category of approaches for achieving transformation invariance is the so-called group-equivariance methods [61–63]. They utilize prior knowledge about the transformation-invariance characteristics of symmetry groups to encode equivariant representations for a combination of different kinds of geometric transformations.

Generalized Geometric Transformations

As already mentioned, generalized transformation-invariant techniques are not restricted to single contexts. They can be further divided into two categories—those that handle arbitrary transformations and those designed to tackle affine transformations. Table 1 summarizes the common techniques employed in handling image transformations in general settings.

Arbitrary Transformations

Many approaches for learning arbitrary geometric transformations generally provide mechanisms for adaptation of receptive fields in such a way that enables filters to capture detailed object variations, including different scales and irregular shapes. These methods are generally highly efficient as no intermediate pre-processing of image data is required to encode invariant features. In [58], for example, Dai, et al. proposed Deformable Convolutional Network (DCN), a CNN-based model that incorporates special modules to learn and apply 2D offsets to the standard (i.e.,

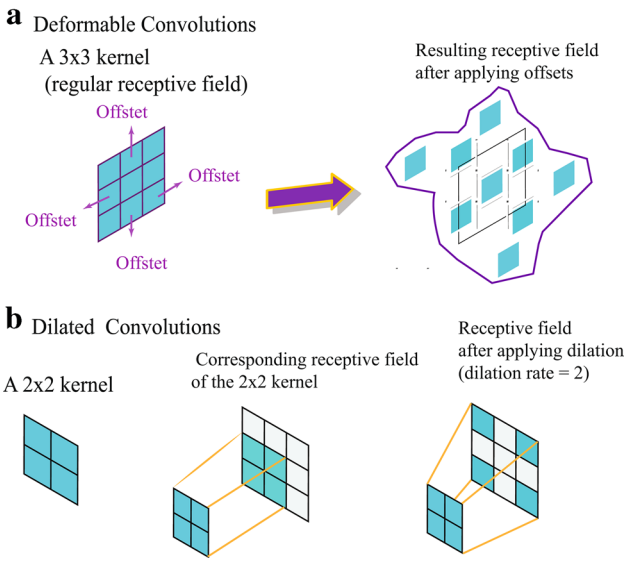

Fig. 6 Many approaches improve invariant generalization by controlling the receptive fteld. Deformable convolution network [89] employ arbitrary offsets learned from input data to adapt the receptive fteld to irregular shapes (**a**). Dilated convolutions [73] (**b**) involve exponentially expanding the fteld of view to capture more visual information without corresponding increase in computational cost

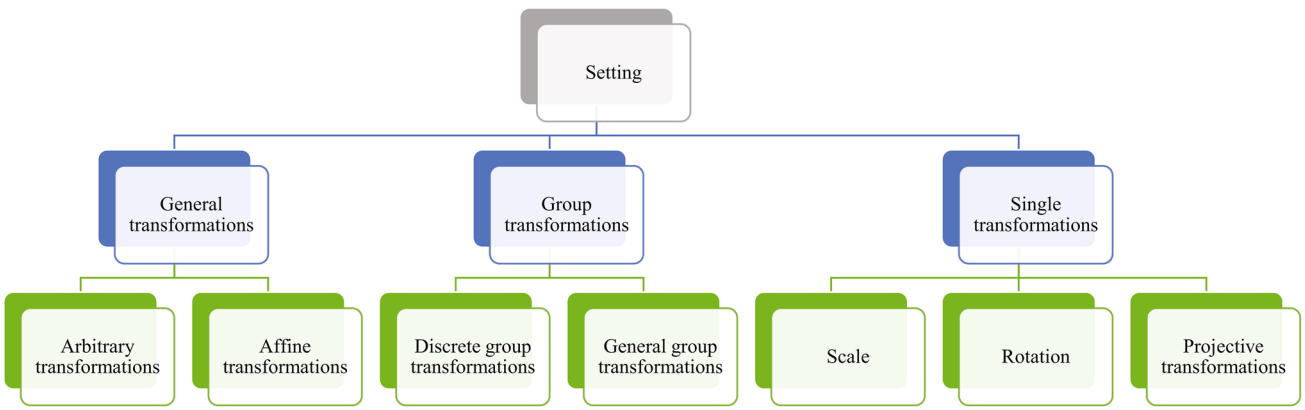

Fig. 5 Taxonomy of geometric transformations as presented in this section

regular) convolutional sampling grids (see Fig. 6). In the approach, the authors introduced an additional layer each into the convolutional and region-of-interest (RoI) layers of standard CNN object detection models based on R-CNN architecture [69]. These new convolution layers were named deformable convolution and deformable RoI layers. As opposed to the fixed bins used in conventional R-CNNs, the deformable convolution and deformable RoI layers employ adaptive 2D offsets to model effects of geometric transformations. These offsets are learned automatically from input image data. The learned offsets, in turn, automatically adjust both the spatial locations and sizes of receptive fields in the deformable convolution and RoI layers according to image scale and shape changes. The weights and learning rate of these additional deformable and RoI connections are set independent of the standard ones. This decoupling has been done to ensure that the deformable connections can easily be incorporated into standard CNNs. Other notable approaches that exploit flexible filter configurations to model arbitrary transformation invariance include Active Convolution [71], Dynamic Filter Network (DFN) [59], Atrous Convolutions [72], also called Dilated Convolution Networks [73]. In [71], Jeon and Kim proposed a flexible convolution technique using what they called Active Convolution Unit (ACU), whose receptive fields are defined by synaptic position parameters and can assume variable shape through training. Dynamic Filter Network [59] employs a dynamic filter generator that automatically generates CNN filters based on input image characteristics. The dynamically generated filters are in turn applied on input images in a location-specific manner. The adaptability of filters allows different spatial configurations of input images to be learned. In [73], Yu and Koltun proposed a new mechanism that introduces spacing, defined by a so-called dilation rate (also known as atrous rate), between elements of convolution kernels. Performing convolution with dilated or atrous filters allows more coverage compared to conventional convolutions. More importantly, the approach allows the effective receptive field to be increased exponentially while maintaining a linear increase in the number of network parameters.

An increasingly popular approach to generalize invariance to arbitrary settings is based on the concept of capsule network or CapsNet. Instead of modeling deformations and other transformations by adapting the receptive field, capsule networks utilize nested multi-layered neural topologies consisting of independent groups of neurons called capsules. In this structure, capsules represent the essence of specific features. In addition, the approach uses activity vectors to encode so-called instantiation parameters which describe extrinsic object properties such as pose, orientation, skew, deformation and scale. In the network, the activity vectors define the probability of the existence of these features. The basic principle of the capsule network was first introduced in 2011 by Hinton et al. [74], and subsequently refined by Sabour et al. [57]. In addition, in [57], an efficient method for training capsule networks, known as routing by agreement, was proposed as a replacement of the standard CNN methods used in previous capsule network implementations. In the approach, predictions about the existence of particular features in the deeper layers rely on consensus of predictions among the shallower (earlier) layers of the network. To further improve the performance of capsule networks, the EM routing algorithm [75], a more effective training technique based on expectation maximization was proposed. The effectiveness of the approach has been widely demonstrated in text classification tasks [76]. Over the past few years, many modifications and extensions of the original ideas developed in [74] and [57] have been suggested [77–81] to further increase the robustness of capsule networks to image transformations and extend their application to more challenging domains, including image recognition under occlusion [81], geometric transformation occurring in visual SLAM and object detection in aerial surveillance [82, 83], road sign recognition in autonomous driving systems [84], semantic scene segmentation [85] and action recognition [86]. The approaches have also been extended to 3D recognition tasks [87, 88].

The main advantage of capsule networks over traditional CNN models is that they are able to learn and preserve spatial relationships that characterize real-world objects. That is, capsule networks actually achieve equivariance—a more useful concept for visual understanding [57]. Capsule networks are also more sample-efficient compared to conventional CNNs. In this regard, the use of capsule networks can significantly reduce overfitting problems and improve recognition accuracy in data-scarce domains. However, as discussed in [57], capsule networks usually attempt to learn every available detail in the input space, resulting in poor performance in highly noisy or cluttered environments. Moreover, since the final outputs of capsules are fused at the deepest layers, their ability to encode local, low-level transformations is severely constrained [89].

Another common approach to learning arbitrary geometric transformation invariance is to extract discriminative mid-level features from lower convolutional layers that describe more basic visual elements in the input image and are invariant to its transformations (Fig. 7). CNN approaches to extracting mid-level features tend to learn visual concepts that resemble deformable part models [91–93]. Recent methods [94–98] employ deep neural networks to extract features and achieve impressive results. The overall goal of these networks is to obtain more diverse discriminative visual features that are robust to various forms of image transformations by combining both mid-level and high-level visual concepts from multiple layers of CNN. The approaches typically employ

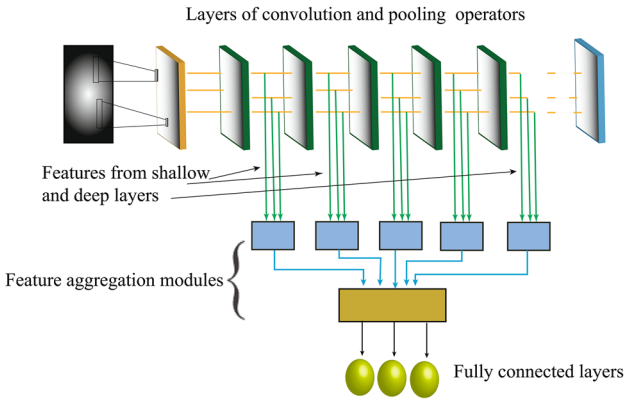

Fig. 7 CNN architecture for deeply learning robust part representations. Geometric transformation-invariant features can be extracted by combining low-level visual concepts from shallower layers with high-level features of deeper layers

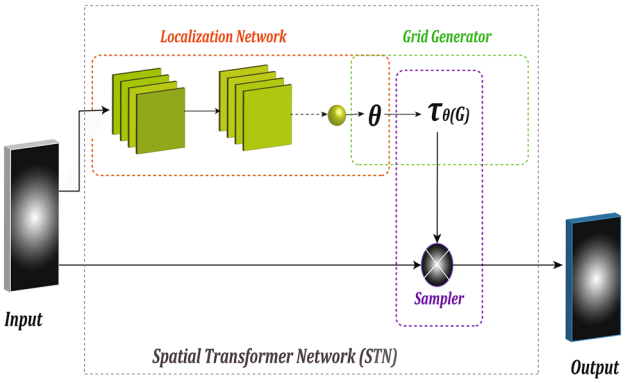

Fig. 8 General structure of the Spatial Transformer Network (STN) [89]. At the heart of the STN are the Grid Generator, Sampler and Localization Network. It generates a sampling meshgrid corresponding to the input image, which is warped by the transformation matrix and sampled. The Sampler interpolates the values of the resulting output feature vector to integer pixel values. The Localization Network learns and generates the appropriate transformation parameters for input images based on the loss propagated from the meshgrid sampler

so-called part filters that are optimized jointly with CNN classifiers to select robust mid-level visual features. In [96], Kortylewski et al. trained a conventional deep CNN whose predictions are augmented with deeply mined mid-level features from the same network. Yang et al. [95] proposed a method for extracting invariant mid-level features based on the use of special feature extraction modules, known as P-CNN, that are embedded in designated layers of pre-trained CNN models to extract useful features. Similarly, Sun et al. [97] proposed an end-to-end deep CNN that incorporates part-level convolutional filter—part-based convolutional baseline (PCB)—to extract mid-level visual concepts directly from input images. The concept of mining features from CNN layers provides benefits in terms of reduced model complexity and increased computational efficiency. Deep learning models that extract mid-level representations directly from CNN feature maps have demonstrated effectiveness in challenging machine vision tasks such as object detection [100], pose estimation [99], human activity recognition [98], person re-identification [97] and semantic scene segmentation [100].

Models that provide invariance to arbitrary geometric transformations are able to learn complex image patterns associated with various real-world phenomena, including non-regular shapes associated with occlusion and deformations. These approaches are crucial for the realization of generic machine vision capabilities. While it is still not possible to fully encode invariance with respect to all geometric transformations using a single network, the approaches described in this subsection provide partial invariance for various kinds of transformations that can adequately solve narrower problems within a variety of acceptable limits. Nevertheless, there is an important argument [101–103] in favor of less general solutions that are designed to tackle more specific invariances in narrower application contexts. One major benefit of this approach is the ease of incorporating prior knowledge about the underlying task into the machine vision pipeline, resulting in guaranteed performance improvement.

Affine Transformation

A number of approaches [60, 89, 104, 105], instead of attempting to learn arbitrary geometric transformations, encode affine transformations. These approaches exploit domain knowledge and utilize approaches based on first principles to explicitly encode transformation invariance. In contrast to most of the techniques described earlier in "Arbitrary Transformations" (specifically, in [59, 71, 72], which rely on warping convolution filters, affine transformation-invariant approaches such as [60, 89, 104, 105] directly transform input features with the help of dedicated modules within CNN layers in such a way as to guide the CNN to learn transformations on the input images. Encoding affine transformations in this way has a number of advantages. First, the number of learnable parameters that are needed to encode invariance reduces significantly. In addition, the requirements for structural complexity to encode transformations also decreases. More importantly, the invariance of the resulting network model to known transformations is guaranteed. One major limitation of this approach, however, is that it requires prior knowledge of the specific transformation to be dealt with. This is not often possible in many situations.

Arguably, the most popular affine transform-invariant CNN architecture is spatial transformer network (STN) [89]. The basic structure of the spatial transformer network is depicted in Fig. 8. It provides an analytical mechanism to explicitly perform different geometric transformations on feature maps or input images in an intermediate pipeline before employing additional CNN layers for feature extraction and classification. The goal is to align feature maps or images to their canonical form. The essence of the approach is to learn the relationship between different image appearances and the underlying geometric transformations. Whereas affine-invariant approaches like the spatial transformer network are not as general as approaches using capsule networks or deformable convolutions, they can provide better solutions in those applications, where more deterministic performance guarantees are required. The original method proposed in [89] learns three types of transformations, namely, affine transformation (scale, rotation, shear and translation); projective transformation and Thin Plate Spline Transformation (TPST). In principle, additional transformations can be introduced to account for a variety of scenarios. Indeed, architectures incorporating additional transformations such as deformation [105] have been suggested. A spatial transformer can be integrated seamlessly into conventional CNN models and trained end-to-end in the normal way through standard training methods like backpropagation. Spatial Transformer Network architectures have shown promising prospects in challenging machine vision tasks, including generic object detection [106], action recognition [107], 2D to 3D scene reconstruction [108] and pose estimation [109]. A major advantage of the method lies in the fact that it is based on analytical techniques that are transparent and function by understandable principles of operation, making end applications easy to debug. However, this transparency is achieved at the expense of higher computational complexity. To overcome the computational burden involved in computing parameters for geometric transformations in approaches such as [89], Tarasiuk and Pryczek [60] proposed to replace complex exponential and trigonometric computations with linear matrix operations. In addition, they proposed to cache the computed matrix coefficients for subsequent use. However, the approach is restricted to situations, where the input and output parameters of the transformation matrix are fixed. For this reason, the approach has very limited scope of application. In [110], Lin and Lucey proposed a modified STN model, known as Compositional Spatial Transformer Networks (IC-STNs) that propagates transformed image parameters instead of image features as in the classical STN. The propagation of parameters reduces the amount of computations needed to encode features, thereby significantly improving efficiency. Freifeld et al. [111] encode transformation-invarance with the help of Continuous Piecewise-Affine (CPA) velocity fields.

Another interesting approach for affine transformation invariance is Transform-Invariant Convolutional Neural Network (TICNN) [104]. It introduces a module in which random scaling, rotations and translations are performed on feature maps. The idea is to discourage the network from tying input images to specific configurations by inducing diversity of feature map topologies through affine transformations. The overall result is robust feature representation that is independent of specific image transformations. Approaches that automatically learn affine transformations have also been proposed [112, 113]. In [112], for example, Wei et al. proposed an end-to-end, deeply learned affine transformation-invariant representation approach using interpolation technique to expand and contract feature maps. They introduced two sub-modules—inflation and interpolation layers—that can be embedded in a deep CNN to automatically transform receptive fields. The transformation parameters are entirely learned from input data without any manual configuration.

In some cases, it is preferable to learn affine transformations using a so-called multi-column or multi-stream network consisting of a set of independent models that learn single transformations instead of a large monolithic model (Fig. 9).

It is important to differentiate this approaches from the methods [60, 89, 104] discussed in the preceding paragraph,

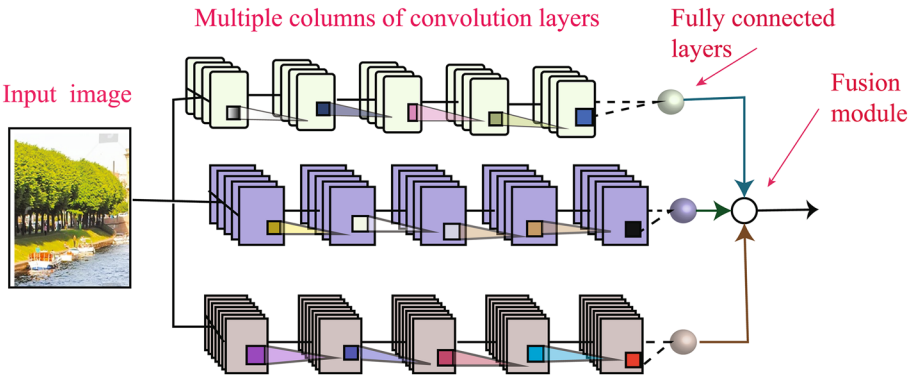

Fig. 9 Simplifted structure of a Multi-column Convolutional Neural Network (MC-CNN). With this conftguration, different transformations are learned independently by each of the constituent branches of the network. Predictions are aggregated using a fusion sub-module

where spatial transformation elements are embedded in a linear cascading manner. With multi-column network architectures [114–116], multiple dedicated CNN sub-networks are organized in columns (i.e., parallel streams), where each column learns a unique transformation or sets of transformations. The basic idea is to use simple parallel models to synergistically learn complex transformations. The use of such architectures in solving transformation-invariant recognition problems is widespread, and new techniques [43, 117–119] are actively being developed by a large number of researchers. The concept was used in the original work by Ciregan et al. [114]. In the approach, the authors proposed to use 35 distinct columns to classify Chinese characters. These columns are trained on input images pre-processed by applying different transformations such as scaling, rotations and translations images. The output activations of the various columns are then averaged to produce a common prediction. Performance analysis by the authors confirmed that the proposed approach outperformed all previously reported methods by about 1.5–5%.

In the multi-column approach described above, image transformation is typically obtained by manual pre-processing but this imposes many constraints in practical situations. In [118] Zhang et al. proposed a Multi-column Spatial Transformer Convolutional Neural Network (MC-STCNN) for traffic sign classification. The basic structure is similar to the above approach, except that instead of using preprocessed images they use a Spatial Transformer Network (STN) [89] to transform image shapes. The network also employs a special module—a so-called Distributer—that scales the input images to five different dimensions before transformation by the spatial transformers. In [43], the authors proposed a different approach for obtaining image transformations without the use of data augmentation. In the approach, input images are transformed by predefined affine transformation functions before passing through designated network branches consisting of convolutional and subsampling layers. The output feature maps are then aggregated and sampled with the help of a specially designed max-pooling technique, TI-Pooling, to encode transformation invariance. A similar approach based on the use of analytical transformer was proposed in [119]. The authors proposed a more general method of learning transformation invariance based on the concept of random image transformations and special feature aggregation module known as drop-transformation-out.

It is worth noting that a number of advanced fusion methods have been proposed to enhance feature aggregation in multi-column networks. These include adaptive weight-learning [120], statistical [121] and probabilistic [122] techniques. Other notable examples are weighted voting [123], blending [124] and meta-combining [125]. Despite the high potential of multi-column architectures for achieving high degree of geometric transformation invariance, recognition approaches based on these techniques have a number of serious drawbacks: (1) using multiple sub-networks can be computationally expensive and slow; (2) the constituent sub-models can be difficult to optimize jointly; and (3) the approach restricts sharing of learned knowledge within the network structure. In addition, in a lot of practical situations image transformations are combinatorial in nature, that is, several transformations occur simultaneously in different permutations—and simple combination of distinct transformations may not suffice.

Despite these limitations, multi-column approaches are in many ways superior to monolithic ones in terms of their modularity and reliability. The network topology is simple, and can be extended to a wide variety of contexts, including arbitrary geometric as well as photometric transformations. Moreover, as machine vision techniques mature and research focus shifts from black box approaches towards the realization of explainable and causal visual recognition systems, developing compact models with well-defined principles of operation to encode different aspects of object interactions will likely become more important.

Single Geometric Transformations

As opposed to the approaches described earlier in "Generalized Geometric Transformations" that simultaneously tackle many kinds of affine and arbitrary transformations, a large number of techniques have been devised to address single geometric transformations separately. The main motivation for such approaches is based on the fact that only a few geometric transformations–translations, rotations and scale variations—dominate in many practical scenarios [64]. Moreover, approaches that tackle these single transformations are generally more efficient and simpler than their generic counterparts. They are, therefore, widely used to improve recognition performance in pre-trained CNN models. Single transformation methods are commonly used in domains, where specific image transformations are expected. For instance, rotation invariance is generally very useful for machine vision tasks in applications like aerial surveillance because of the different orientations from which cameras would usually capture images. Projective transformations are common application domains like autonomous driving and augmented reality, where wide field of view (FoV) image sensors are usually used to capture 360 degree or panoramic images.

It is largely accepted that deep CNNs are already invariant to translations [30, 89]. Consequently, rotation and scale transformations are the predominant problems these class of approaches commonly tackle. Typically, the techniques exploit various strategies, including analytical

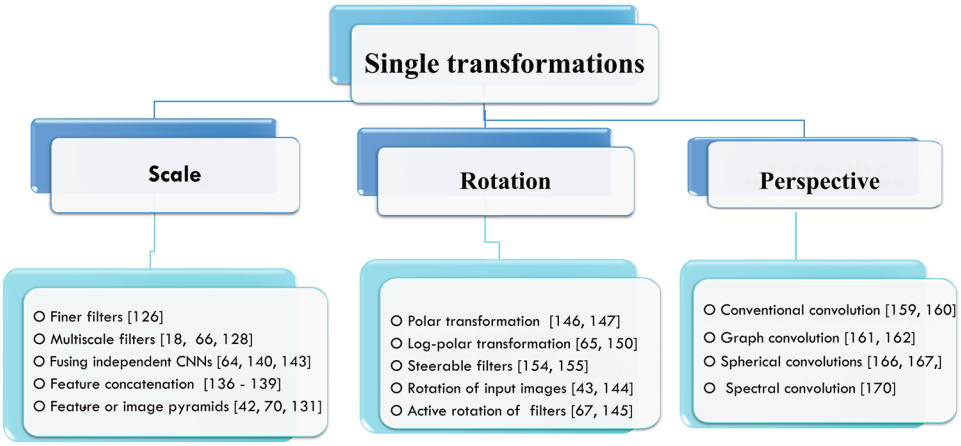

Fig. 10 Common approaches to tackling single transformations

preprocessing, transformation parameters, and topological correlation between transformed image features and convolution filter configurations or feature maps to learn invariance or equivariance. In Fig. 10, we summarize some of the most important approaches for tackling single transformations.

Scale Invariance

In CNNs, filters of smaller dimensions generally learn finer image features while large-size filters capture larger, more global or higher level features. Consequently, the selection of filter dimensions invariably takes into account the granularity of features that needs to be learned. For instance, Simonyan and Zisserman [126] employs a network topology that uses small-size (1×1) filters to enhance the network's ability to extract fine-grained features. However, since the kernel size is fixed, techniques based on these approaches only provide contextual details relevant for fine image features and do not capture course details or multi-scale information inherent in real-world settings. Based on the correlation of filter dimension and the granularity of the generated features, approaches to scale invariance commonly utilize multi-scale filters, that is, combinations of different filter sizes, to handle different scales [18, 66, 127, 128]. For instance, Szegedy et al. [127], employ small blocks of convolutional elements consisting of differently sized kernels—specifically, 5×5, 3×3 and 1×1 kernels—to enable feature extraction at different scales. In [129, 130], a new concept is proposed that employs competitive pooling strategy based on maxout activation to replace the conventional feature aggregation pooling methods [127] used for multi-scale filters.

An alternative and highly popular approach, known as filter pyramid network, adopts a spatial filtering scheme that employ a pyramidal structure [42, 70, 131] of convolutional filters to extract varying sizes of features. Lin et al. [70] proposed an approach that exploits CNN's inherent pyramidal feature hierarchy to encode scale invariance without the need for creating extra multi-scale feature maps or images. From a single image, the method generates multiple size feature maps at different levels of the CNN pipeline. In [131], Chen et al. introduced Scale Pyramid Network (SPN) which utilizes a specialized module to generate multi-scale pyramid of features using different dilated convolution rates in parallel within deeper CNN layers. Many standard object detection models [132–134] employ multi-scale images in pyramidal structure during training to ensure scale invariance. Image Pyramid networks typically scales input images into different sizes and then train multiple independent sub-networks on each scale. In contrast to methods such as [18, 128] which employ large networks with multi-scale filters, approaches such as [64, 135] utilize multiple CNN sub-models, each with its own filters for extracting features of a particular scale. Van Noord and Postma [64], for example, proposed a scale-invariant model that combines four different sub-models designed to handle different scales into a composite assembly. Individual predictions from the separate CNNs are averaged to produce a final prediction.

Rather than using different filter sizes or employing separate sub-models to handle different scales, some works [136–139] concatenate features from different layers of the network hierarchy. The idea is to leverage larger structural representations from higher layers together with finer geometric details captured by lower layers for scale-invariant prediction. Other approaches [140–143] have been proposed based on multi-branch network architectures in which different layers independently perform predictions appropriate to the object scale. Predictions across the different layers are then averaged to produce the overall prediction. Some of the common CNN architectures that encode scale invariance are shown in Fig. 11.

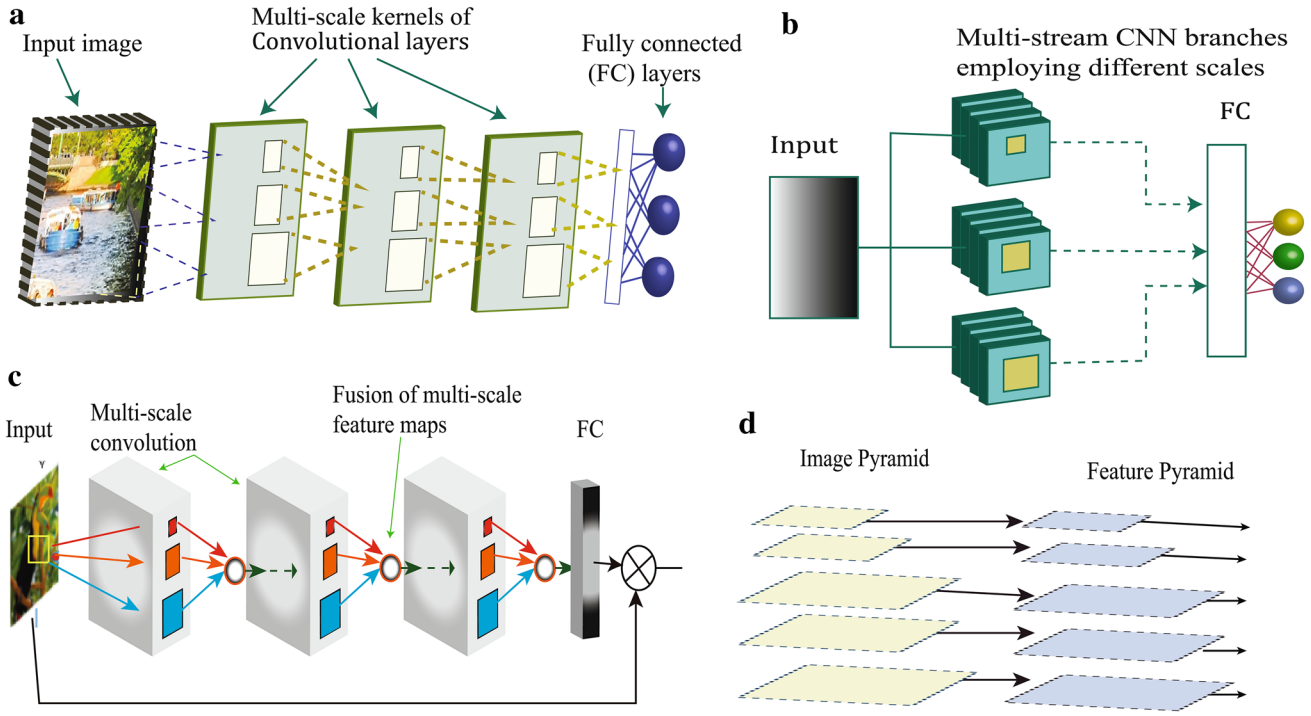

Fig. 11 Strategies for encoding scale invariance include using different filter sizes in each convolution layer (**a**), employing independent sub-networks for learning different scales and then aggregating their output predictions (**b**), using multi-scale ftlters with competitive pooling to select best option (**c**), and exploiting image and/or feature pyramids (**d**)

Rotation Invariance

A variety of approaches have been exploited to ensure rotation invariance in deep CNNs. The simplest strategy is to encode pre-determined, multiple rotated positions. This can be done by performing successive rotation operations on input images [43, 144] or convolution filters [63, 67, 145]. In [63] Marcos et al. applied successive rotation operations to the convolution filters in discrete steps. They then extracted vector field feature maps by spatially convolving the rotated filters on input images. Oriented Response Networks [68] introduces Active Rotating Filters (ARFs) that are discretely rotated in the process of convolution to generate feature vectors with corresponding location and orientation information.

An alternative approach is to transform the image domain into a new domain in which angular translations or rotations become linear translations. For this purpose, various methods have been suggested, including polar canonical coordinate transforms [146–149]. These techniques convert rotation of an image in the original Cartesian coordinate system into translation in a polar coordinate system by interpolating pixel values of the image onto corresponding locations of a planer grid (Fig. 12a). Jiang and Mei [146] proposed a rotation-invariant CNN with a dedicated polar transformation layer that can be inserted into to learn rotation invariance.

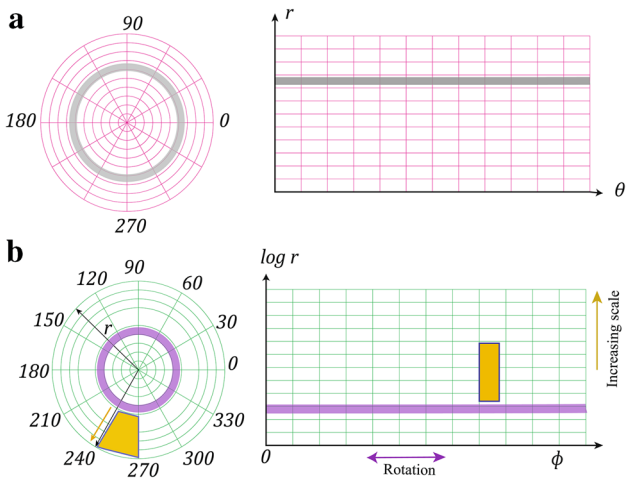

Fig. 12 Polar (**a**) and log-polar (**b**) representations transform rotation in Cartesian coordinate system to linear translations. Scale changes is equivalent to translation in the vertical axis of the log-polar plane

Similarly, Polar Transformer Network (PTN) [149] incorporates a polar transformer unit to transform features to polar canonical coordinate representations. In [147], Chen et al. proposed two different polar transformation modules, Full Polar Convolution (FPolarConv) and Local Polar

Convolution (LPolarConv), each of which can independently encode rotation-invariant representations.

Another type of domain conversion is based on log-radial harmonics or log-polar representation [65, 150–153], which renders the rotation of an image in the Cartesian coordinate system as a plane translation in one axis and scale change as a translation along the other main axes of the logarithmic-polar plane. A key advantage of the polar-log transformations lies in the fact that both rotation and scale are encoded as opposed to rotation-only invariance in the case of the polar transform approach. However, the polar transform has the advantage of simplicity and lower computational cost. With the log-polar transformation, images are projected from Cartesian coordinate into logarithmic-polar coordinate representation (Fig. 12b). As depicted in Fig. 12b, this approach converts the rotation of an image in Cartesian coordinate system into a linear translation along the horizontal axis and at the same time encodes scale change as a corresponding translation along the vertical axis.

A major drawback of these coordinate transformation approaches, however, is that they invariably introduce unwanted noise and distortions as a result of image pixels at the central parts of the image being sampled more aggressively than those at its ends. Another significant limitation of the techniques discussed is that they are limited to a finite set of discrete rotations. This problem can be eliminated using steerable filters [154, 155]—a special class of filters capable of encoding arbitrary rotations in continuous space using linear combinations of simple filters, known as basis filters. Because of their power in representing continuous rotation information, steerable filters are being widely used to tackle more challenging visual recognition problems involving rotation invariance [65, 156–158]. Worrall et al. [90], proposed Harmonic Networks to extend these discrete rotation-invariant techniques to continuous 360 degree rotations using circular harmonics in place of standard convolutional filters. Ghosh and Gupta [65], proposed scale-steerable filters—a type of filter design formulated on the basis of log-radial harmonics—that combines both rotation- and scale-steerable circular harmonics to extend these representations to broader domains involving continuous rotations.

Projective Transformation

Many machine vision applications utilize wide field-of-view (FoV) or omnidirectional cameras so as to capture large amounts of visual information in the form of spherical 360 degree and panorama views. Meanwhile, wide FoV data sets are currently scarce, leaving CNN models to increasingly rely on perspective images for training. To make predictions in spherical image domains, some approaches first convert the input spherical images into canonical views before extracting useful features. In [159], for examples, spherical image data is projected onto 2D planes, where standard convolution is applied on the converted planar images. The main limitation of this approach is the enormous computational cost in converting from spherical to 2D planar views. In addition, the conversion introduces errors, making standard convolutions less effective. To mitigate both problems, Monroy et al. [160] proposed to divide omnidirectional input images into equally sized patches and then map these patches onto planer surfaces (specifically, six faces of a virtual cube). With this representation, a conventional CNN can then be used to extract features for inference. Some approaches (e.g., [161]) have proposed to treat spherical images as graph-structured data and use so-called graph convolutions to extract invariant image features. To further enhance invariance, Khasanova and Frossard [162] additionally modeled into the graph structure prior knowledge about the camera lens geometry. While these works have proven effective in tackling invariant recognition in spherical image domains, due to the complexity of the representations, they are less general and much less powerful compared to standard CNNs.

A number of works [163–167] proposed special convolutions to directly extract transformation-invariant features from omnidirectional images. In [163], Cohen et al. introduced a new type of convolution, known as Spherical convolution, where the sliding of the convolution filter translates to a rotation on sphere. This ultimately makes the approaches equivariant to rotation of objects on the sphere. Similarly, Boomsma et al. in [167] applied concentric cubed-sphere convolution on spherical data represented in cubed-sphere form. Spherical CNNs [165-169] typically interpolate features in spherical images onto 2D plane by equirectangular projection using Fast Fourier Transform (FFT) technique to compute spherical cross-correlations of image features. However, the conversion of spherical images into planer projections inevitably introduces perspective distortions and decreases performance. To mitigate this problem, some researchers [163, 170] have proposed to first transform convolutional filters and feature maps into spectral representations before performing convolution. In [171], Su and Grauman proposed to directly transfer conventional convolution filters to spherical 360 degree images by utilizing a dedicated learnable function to handle spherical to planer image transformations. Spherical networks have been applied to handle images produced by omnidirectional image sensor systems [168]. This is particularly relevant in machine vision applications such as autonomous driving, where the use of wide field-of-view image sensors enhances situational awareness. They are also useful in augmented reality applications [172] and in aerial imagery [166], where they are used to interpret rotated, non-canonical image views generated by wide field-of-view (FoV) cameras. Their application has also been extended to non-Euclidean domains,

for example, for interpreting cosmological maps [173] and molecular data analysis [167].

Group Transformations

Basic Concepts of Symmetry Group Equivariance

A large part of the success of CNNs in visual recognition tasks is due to the equivariance of the convolution operation to translations. Recently, much research effort has been dedicated to extending this equivariance property to other geometric transformations. An increasingly popular class of geometric transformation-invariant recognition approaches [61, 62, 149, 174, 189, 190] exploit insights from group theory [175] to make CNN models equivariant to group or symmetry transformations—that is, the set of geometric transformations that leave the semantic meaning of the underlying image unchanged with respect to a given context. Specifically, group transformation G has the property that features of an image transformed by G are the same as the original image under the action of the group transformation G. More formally, given a feature representation f of image i under group transformation $G, f(G(i)) = G(f(i))$. Group equivariant networks make use of these principles to allow models to mathematically describe a wide class of image domains using systems of functions whose values are constant under the influence of group transformations. Obviously, such a definition of the invariant recognition problem limits the scope of these approaches. The main difficulty when applying the group-equivariant methods in practical situations is the fact that most real-world problems are usually very complex; many of them cannot be formulated in precise terms. In many cases (e.g., [176, 177]), group-equivariant models incorporate diverse techniques for encoding invariants, with only one of which, in reality, exploiting group theory as the basis of its operation. However, it possible to apply CNNs built on the basis of symmetry group principles alone to invariant recognition visual recognition problems. In specific contexts, the approaches have demonstrated good results in practice [61]. Symmetry group equivariant networks can loosely be classified [178] into those that tackle specific (single) transformations (e.g., rotations) or a compact set of discrete transformations (e.g., a combination of rotations and reflections) and those that guarantee equivariance for general and arbitrary transformations. Some of the common methods describe below are captured in Table 2.

Single and Locally Compact Symmetry Transformations Over Discrete Space

Approaches that exploit symmetry constraints have been proposed to deal with specific transformations such as scale, rotations and projective transformations. Scale-equivariant networks [181–184] generally exploit scale-space theory and semigroup properties to achieve equivariance to scale transformations. In many of these works, the goal is to encode multi-scale features while preserving translation-equivariance. Rotation-equivariant approaches based on rotating convolutional filters or feature maps are proposed in [62, 63, 185, 186]. These approaches typically require explicit rotation operations to be performed while exploiting cyclic symmetry constraints. A large number of works have also been proposed to handle spherical to planer projective transformations [187, 188]. An important property of symmetry groups is that an image that is equivariant under a composite symmetry transformation will remain equivariant under the sequential application of the constituent transformations [189]. In other words, complex image transformations can be encoded using a combination of elementary transformations. Consequently, complex geometric transformation-invariant recognition tasks can be reduced to finite sets of simple image transformations in a discrete space and can be encoded by predefined geometric transformations. This property allows group equivariant techniques to leverage prior knowledge of the properties of symmetry groups [61] to simultaneously provide a compact equivariant representation with respect to multiple transformations. In their original paper [61], Cohen and Welling introduced the concept of G-CNN, a CNN architectures that incorporates so-called group convolutions (or G-convolutions). The idea proposed in [61] was to model a combination of

Table 2 Common approaches to tackling symmetry group transformations

Target transformations	Key approaches	References
Discrete symmetry group transformations	Scale-space theory and semigroup techniques	[181–184]
	Manipulation (e.g., rotation) of feature maps or filters	[63, 185, 186]
	Projective transformations	[187, 188]
General group transformations	B-spline interpolation methods	[195, 196]
	Capsule network-based approaches	[179]
	PDE-based approaches	[190]
	Steerable filters	[155]

90° rotations and mirror reflections that can be learned by CNN filters through symmetry group action. The results of their study demonstrated a superior performance over conventional CNNs trained and tested on CIFAR10 and rotated MNIST data sets. Group equivariant networks such as [61, 62, 178] are limited to a small, fixed number of discrete transformations. For example, in [61], only a few transformations—specifically, 90° rotations and mirror reflections—are possible.

General Symmetry Transformations Over Continuous Space

Recently, a number of techniques have been proposed to generalize symmetry group equivariance to arbitrary transformations and over continuous space [150, 174, 175, 190–193]. In [190], the authors proposed a partial differential equation (PDE) approach to extend equivariance to general settings. The technique employed was to treat convolutional network layers as partial differential equation solvers. The neural network provides generalized equivariant representation by means of linear and morphological convolutions using these partial differential equations solvers. Henriques and Veldhadi [150] introduced a new convolution layer, called warp convolution layer, where input images are transformed by exponential maps before being convolved by standard convolution operations. In the implementation, they proposed to use bilinear resampling to generate efficient convolutions that are amenable to continuous transformations. Another technique for constructing general equivariant convolutional networks over continuous rotations is based on the concept of filter steerability [155]. This approach utilizes steerable filters in place of conventional CNN filters to learn equivariant representations of input images. In [175], Weiler and Cesa proposed a general method for constructing invariants, and a concept to extend approaches dealing with specific symmetry transformations (e.g., [63, 90, 155, 194]) but whose representations lend themselves well to generalization.

Methods based on B-spline interpolation [195, 196] are also being used to achieve equivariance over continuous transformations. In [195], Bekkers proposed to generalize arbitrary symmetry patterns over continuous transformations using B-spline basis functions for representing group convolution kernels. Approaches that extend existing CNN representation concepts to group equivariant networks have been widely studied. In [62], the authors modeled rotation equivariant priors in the initial layers of CNN and propose a new training technique based on what they called Soft Rotation-Equivariant CNN that encourages equivariance on training samples.

Romero et al. [180] proposed Attentive Group Convolutions, an approach inspired by biological visual attention mechanism [197], which allows group equivariant networks to learn meaningful relationships among different symmetry transformations. Lenssen et al. [179] extended the concept of group equivariance to capsule networks. The approach implements capsule networks' routing by agreement algorithm on symmetry groups. Lately, CNN models based on the principles of symmetry group equivariance have also been extended to Generative Adversarial Networks [198].

Emerging Trends and Future Research Directions

Modern deep learning models are becoming more complex and often highly specialized for very narrow tasks. An emerging trend is to build large, heterogeneous deep learning models that employ elementary sub-models to handle specific sub-problems. In this regard, it is conceivable that many future approaches could utilize composite CNN architectures consisting of diverse models specialized in dealing with specific geometric transformations. For example, transformations such as rotation, scale, deformation and skew could be handled by separate models within a larger network such as the architecture depicted in Fig. 9. Models based on this concept would generally be more interpretable and offer better deterministic performance guarantees. In particular, they may capture more nuanced details of the underlying scenario, making them less susceptible to fooling and adversarial attacks as compared to large, homogeneous neural network models. Given all of these attractive characteristics of hybrid architectures as models for learning complex phenomena such as geometric image transformations, one can reasonably expect the development in this direction to accelerate as these integrated models leverage rapidly expanding workload-intensive technologies such as cloud computing and high-performance computing (HPC) resources. Moreover, new machine learning concepts such as knowledge amalgamation [199], domain generalization [200] and meta-learning techniques [201] can be leveraged to develop more general CNN architectures that are invariant to complex image transformations. An increasingly important aspect of future research work will, therefore, be the development of sophisticated, multi-modal techniques that are not only robust but are also at the same time more general and provide various degrees of adaptation to different image transformation in diverse scenarios. At the moment such recognition techniques have very narrow scope, being trained for specific tasks each time; they are currently very rigid and do not scale well. As the approaches develop further, it will be possible to build multi-purpose models to tackle generic machine vision problems.

Also interesting are research works associated with developing biologically inspired deep learning techniques. Of particular importance are artificial neural networks using feedback connections [202]. These methods enable the realization in artificial networks important functional properties of biological visual systems—notably, memory and attention mechanism [197, 203]—that are pivotal in building neural network architectures for efficient and robust visual recognition. Just as in biological vision, attention mechanism in deep learning models enables representation methods that capture only the most relevant visual cues while ignoring unnecessary information, resulting in more efficient and robust representation. Models employing memory will allow artificial neural networks to retain previously learned features while incorporating new information about image transformations from subsequent training sets. Biologically inspired machine vision techniques are already playing an important role in the development of robust visual representations for challenging tasks such as few-shot learning [204]. As these approaches continue to evolve, an interesting prospect will be their integration into CNN models to facilitate general transformation-invariant visual understanding.

Perhaps, the most promising direction for future work is in the area of Automated Machine Learning (AutoML) [205, 206], the development of algorithms to automatically generate machine learning models for different tasks. A particularly interesting aspect of these works, Neural Architecture Search (NAS) [207], a subset of AutoML, specifically focuses on automating neural network architecture design. This entails using machine learning algorithms to iteratively build and test several different architectures from a given data set and task, and then using various search strategies, to select the best performing model from the generated candidates. Techniques based on NAS have already produced results better than state-of-the-art hand-crafted neural networks in some machine vision domains [208–211]. However, a major limitation of NAS approaches is their heavy reliance on very large, "complete" training data set to produce "best-fit" neural architectures. Without the ability to model different nuances of real-world settings, it will be challenging to encode complex geometric transformations with NAS generated models—especially in real-world applications, where training data are scarce. In addition, the processes of model generation and search are currently enormously computationally expensive.

Currently, active research areas include the use of meta-learning techniques to improve the sample efficiency of AutoML-based approaches [212, 213], development of better search strategies [142, 143] and computationally efficient techniques for model generation [214, 215]. In the foreseeable future, however, better NAS algorithms and improvements in hardware performance—especially cloud-based computational resources—will enable the development of NAS techniques that can be used in challenging domains to solve challenging machine vision problems. Indeed, big tech firms are already starting to deliver AutoML based platforms as a commercial services. Key among them include Google Cloud AutoML and Microsoft Custom Vision services. In addition to this commercial products, open source development tools, for example, Auto-keras [216] and Auto-sklearn [217], have also been introduced. With these new developments, the prospect for machines generating game-changing architectures fundamentally different from present hand-crafted approaches is very real.

Summary and Conclusion

In this paper, we reviewed recent methods for learning geometric transformation-invariant representations in deep learning models. Geometric transformation here is understood as 2D planer image geometry transformations that result from visual appearance changes of the underlying objects in real-world 3D scenes. The main focus is on approaches that employ classical CNN architectures as baseline for building more sophisticated models for transformation-invariant generalization.

First, we briefly described feature representation and generalization properties of convolutional neural networks, with special focus on robustness to minor image transformations. We then surveyed state-of-the-art techniques that extend the capabilities of classical CNN architectures to handle more aggressive geometric transformations. Although the approaches as presented in this survey have been categorized into three groups based on their universality with respect to the range of geometric transformations they are designed to handle, the underlying principles and implementation details are extremely diverse. The common principles for encoding geometric transformation invariance include: appropriately transforming convolution filters or feature maps, analytically pre-processing input data in an intermediate pipeline, or transforming the image domain into a new domain, where invariance can be easily encoded. In addition, there are approaches in which the invariance of features is provided by the special structure of the neural network as a whole. The most well-known example of this class of approaches is the capsule network. Other notable examples include multi-stream or multi-column architectures and hybrid models that employ a combination of specialized neural network units for encoding specific invariances. The review shows that although an adequate model for generalized geometric transformations (i.e., universal method for all input images under arbitrary geometric transformations) has

not yet been formulated, several techniques exist for solving task-specific problems. Using these methods, it is possible to develop case-based, task-oriented, robust machine vision models to deal with nontrivial geometric image transformations in practical application settings. Despite the outstanding results achieved in recent years, there is still room for significant advances in the near future. New developments in areas like automated machine learning and bio-inspired computer vision methods are expected to drive the next generation of geometric transformation-invariant deep learning architectures.

Acknowledgements The work was entirely done by the authors without the direct involvement of any third party.

Author Contributions The individual contributions of authors to this manuscript is specified below. AM: conceived the study and formulated the original problem. FM: helped in providing the initial direction of the work. Both authors jointly developed the methodology and categorized common approaches for encoding geometric transformation invariance. AM: surveyed approaches to transformation invariance in general contexts (that is, affine and arbitrary transformations). In addition, he analyzed the various methods and architectures for rotation-equivariance, surveyed new research and development trends relating to the design of deep convolutional neural network architectures. FM: principally surveyed approaches for symmetry group equivariance, as well as techniques for encoding invariance in single transformations (specifically, scale, rotation and projective transformations). Both authors jointly design the artwork for the illustrations. The authors also equally participated in drafting the manuscript and its subsequent revisions.

Funding No funding has been received for this work.

Availability of Data and Materials The work does not involve primary data—we report on works that have already been published.

Declarations

Conflict of Interest There are no competing interests or conflict of interests associated with this work, and there has not been any support or external involvement of any sort in this work.

Consent to Participate Not applicable. No person's data have been used in this work. Consequently, consent is not needed for its publication.

Final Comments Both authors agree to be fully responsible for all aspects of the work, including the content and all ethical and legal issues.

References

1. Alcorn MA, Li Q, Gong Z, Wang C, Mai L, Ku WS, et al. Strike (with) a pose: neural networks are easily fooled by strange poses of familiar objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 4845–54.
2. Lenc K, Vedaldi A. Understanding image representations by measuring their equivariance and equivalence. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. pp. 991–99.
3. Fischler MA, Elschlager RA. The representation and matching of pictorial structures. IEEE Trans Comput. 1973;100(1):67–92.
4. Mundy JL. Object recognition in the geometric era: A retrospective. In: Toward category-level object recognition. Springer; 2006. p. 3–28.
5. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, et al. The history began from alexnet: a comprehensive survey on deep learning approaches. arXiv preprint. 2018. arXiv:1803.01164.
6. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p.p 770–78.
7. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Commun ACM. 2017;60(6):84–90.
8. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E. Deep learning for computer vision: a brief review. Comput Intell Neurosci. 2018;2018:1–13. https://doi.org/10.1155/2018/7068349.
9. Fukushima K, Miyake S. Neocognitron: a self- organizing neural network model for a mechanism of visual pattern recognition. In: Competition and cooperation in neural nets. Springer; 1982. p. 267–85.
10. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient- based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
11. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint. 2012. arXiv:1207.0580.
12. Plagianakos V, Magoulas G, Vrahatis M. Learning rate adaptation in stochastic gradient descent. In: Advances in convex analysis and global optimization. Springer; 2001. p. 433–44.
13. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Berlin: ICML; 2010.
14. Scherer D, Müller A, Behnke S. Evaluation of pooling operations in convolutional architectures for object recognition. In: International conference on artificial neural networks. Springer; 2010. p. 92–101.
15. Moody J, Hanson S, Krogh A, Hertz JA. A simple weight decay can improve generalization. Adv Neural Inf Process Syst. 1992;4:950–57.
16. Hubel DH, Wiesel TN. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol. 1962;160(1):106.
17. Riesenhuber M, Poggio T. Hierarchical models of object recognition in cortex. Nat Neurosci. 1999;2(11):1019–25.
18. Gong Y, Wang L, Guo R, Lazebnik S. Multi-scale orderless pooling of deep convolutional activation features. In: European conference on computer vision. Springer; 2014. p. 392–407.
19. Kheradpisheh SR, Ghodrati M, Ganjtabesh M, Masquelier T. Deep networks can resemble human feed-forward vision in invariant object recognition. Sci Rep. 2016;6:32672.
20. Fukushima K. Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw. 1988;1(2):119–30.
21. Xiao YP, Lai YK, Zhang FL, Li C, Gao L. A survey on deep geometry learning: from a representation perspective. Comput Vis Media. 2020;6(2):113–33.
22. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell. 2009;32(9):1627–45.
23. Müller M, Casser V, Lahoud J, Smith N, Ghanem B. Sim4cv: a photo-realistic simulator for computer vision applications. Int J Comput Vis. 2018;126(9):902–19.

24. Roska T, Hamori J, Labos E, Lotz K, Orzo L, Takacs J, et al. The use of CNN models in the subcortical visual pathway. IEEE Trans Circ Syst I. 1993;40(3):182–95.
25. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). IEEE; 2017. pp. 1–6.
26. Zaniolo L, Marques O. On the use of variable stride in convolutional neural networks. Multimedia Tools Appl. 2020;1–18.
27. Murray N, Perronnin F. Generalized max pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. pp. 2473–80.
28. Kuan K, Manek G, Lin J, Fang Y, Chandrasekhar V. Region average pooling for context-aware object detection. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE; 2017. pp. 1347–51.
29. Khan A, Sohail A, Zahoora U, Qureshi AS. A survey of the recent architectures of deep convolutional neural networks. Artif Intell Rev. 2020;53(8):5455–516.
30. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
31. Luo W, Li Y, Urtasun R, Zemel R. Understanding the effective receptive field in deep convolutional neural networks. Adv Neural Inf Process Syst. 2016;29:4898–906.
32. Araujo A, Norris W, Sim J. Computing receptive fields of convolutional neural networks. Distill. 2019;4(11):e21.
33. Montserrat DM, Lin Q, Allebach J, Delp EJ. Training object detection and recognition CNN models using data augmentation. Electron Imaging. 2017;2017(10):27–36.
34. Savalle PA, Tsogkas S, Papandreou G, Kokkinos I. Deformable part models with cnn features. In: Deformable Part Models with CNN Features. European Conference on Computer Vision, Parts and Attributes Workshop, Sep 6, 2014, Zurich, Switzerland (hal-01109290).
35. Tang W, Yu P, Zhou J, Wu Y. Towards a unified compositional model for visual pattern modeling. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. pp. 2784–93.
36. Kortylewski A, He J, Liu Q, Yuille AL. Compositional convolutional neural networks: a deep architecture with innate robustness to partial occlusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 8940–49.
37. Jack D, Maire F, Shirazi S, Eriksson A. IGE- Net: Inverse graphics energy networks for human pose estimation and single-view reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 7075–84.
38. Halder SS, Lalonde JF, Charette Rd. Physics-based rendering for improving robustness to rain. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. pp. 10203–12.
39. Clevert DA, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint. 2015. arXiv:1511.07289.
40. Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30; 2013. p. 3.
41. Goodfellow I, Warde-Farley D, Mirza M, Courville A, Bengio Y. Maxout networks. In: International conference on machine learning. PMLR; 2013. pp. 1319–27.
42. He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell. 2015;37(9):1904–16.
43. Laptev D, Savinov N, Buhmann JM, Pollefeys M. TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp. 289–97.
44. Yu D, Wang H, Chen P, Wei Z. Mixed pooling for convolutional neural networks. In: International conference on rough sets and knowledge technology. Springer; 2014. p. 364–75.
45. Zeiler MD, Fergus R. Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint. 2013. arXiv:1301.3557.
46. Wan L, Zeiler M, Zhang S, Le Cun Y, Fergus R. Regularization of neural networks using dropconnect. In: International conference on machine learning; 2013. pp. 1058–66.
47. Larsson G, Maire M, Shakhnarovich G. Fractalnet: Ultra-deep neural networks without residuals. arXiv preprint. 2016. arXiv:1605.07648.
48. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint. 2015. arXiv:1502.03167.
49. Wei Z, Zhang J, Liu L, Zhu F, Shen F, Zhou Y, et al. Building detail-sensitive semantic segmentation networks with polynomial pooling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 7115–23.
50. Estrach JB, Szlam A, LeCun Y. Signal recovery from pooling representations. In: International conference on machine learning. PMLR; 2014. pp. 307–15.
51. Ouyang W, Luo P, Zeng X, Qiu S, Tian Y, Li H, et al. Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. arXiv preprint. 2014. arXiv:1409.3505.
52. Girshick R. Fast R-CNN object detection with Caffe. Microsoft Res. 2015.
53. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data. 2019;6(1):60.
54. Paulin M, Revaud J, Harchaoui Z, Perronnin F, Schmid C. Transformation pursuit for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. pp. 3646–53.
55. Azulay A, Weiss Y. Why do deep convolutional networks generalize so poorly to small image transformations? arXiv preprint. 2018. arXiv:1805.12177.
56. Engstrom L, Tsipras D, Schmidt L, Madry A. A rotation and a translation suffice: fooling CNNs with simple transformations. arXiv preprint. 2017;1(2):3. arXiv:1712.02779
57. Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: Advances in neural information processing systems; 2017. pp. 3856–66.
58. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, et al. Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision; 2017. pp. 764–73.
59. Jia X, De Brabandere B, Tuytelaars T, Gool LV. Dynamic filter networks. In: Advances in neural information processing systems; 2016. pp. 667–75.
60. Tarasiuk P, Pryczek M. Geometric transformations embedded into convolutional neural networks. J Appl Comput Sci. 2016;24(3):33–48.
61. Cohen T, Welling M. Group equivariant convolutional networks. In: International conference on machine learning; 2016. pp. 2990–9.
62. Dieleman S, De Fauw J, Kavukcuoglu K. Exploiting cyclic symmetry in convolutional neural networks. arXiv preprint. 2016. arXiv:1602.02660.
63. Marcos D, Volpi M, Komodakis N, Tuia D. Rotation equivariant vector field networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. pp. 5048–57.
64. Van Noord N, Postma E. Learning scale-variant and scale-invariant features for deep image classification. Pattern Recogn. 2017;61:583–92.

65. Ghosh R, Gupta AK. Scale steerable filters for locally scale-invariant convolutional neural networks. arXiv preprint. 2019. arXiv:1906.03861.
66. Li J, Liang X, Shen S, Xu T, Feng J, Yan S. Scale- aware fast R-CNN for pedestrian detection. IEEE Trans Multimedia. 2017;20(4):985–96.
67. Marcos D, Volpi M, Tuia D. Learning rotation invariant convolutional filters for texture classification. In: 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE; 2016. pp. 2012–7.
68. Zhou Y, Ye Q, Qiu Q, Jiao J. Oriented response networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 519–28.
69. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. pp. 580–7.
70. Lin TY, Doll´ar P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. pp. 2117–25.
71. Jeon Y, Kim J. Active convolution: learning the shape of convolution for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 4201–9.
72. Chen LC, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv preprint. 2017. arXiv:1706.05587.
73. Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. arXiv preprint. 2015. arXiv:1511.07122.
74. Hinton GE, Krizhevsky A, Wang SD. Transforming auto-encoders. In: International conference on artificial neural networks. Springer; 2011. p. 44–51.
75. Hinton GE, Sabour S, Frosst N. Matrix capsules with EM routing. In: International conference on learning representations; 2018.
76. Zhao W, Ye J, Yang M, Lei Z, Zhang S, Zhao Z. Investigating capsule networks with dynamic routing for text classification. arXiv preprint. 2018. arXiv:1804.00538.
77. Venkatraman S, Balasubramanian S, Sarma RR. Building deep, equivariant capsule networks. arXiv preprint. 2019. arXiv:1908.01300.
78. Phaye SSR, Sikka A, Dhall A, Bathula D. Dense and diverse capsule networks: making the capsules learn better. arXiv preprint. 2018. arXiv:1805.04001.
79. Ramasinghe S, Athuraliya C, Khan SH. A context- aware capsule network for multi-label classification. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 0–0.
80. Zhang L, Edraki M, Qi GJ. Cappronet: Deep feature learning via orthogonal projections onto capsule subspaces. In: Advances in Neural Information Processing Systems; 2018. pp. 5814–23.
81. Rodrıguez-Sanchez A, Dick T. Capsule Networks for Attention Under Occlusion. In: International Conference on Artificial Neural Networks. Springer; 2019. pp. 523–34.
82. Prakash S, Gu G. Simultaneous localization and mapping with depth prediction using capsule networks for uavs. arXiv preprint. 2018. arXiv:1808.05336.
83. Mekhalfi ML, Bejiga MB, Soresina D, Melgani F, Demir B. Capsule networks for object detection in UAV imagery. Remote Sensing. 2019;11(14):1694.
84. Kumar AD. Novel deep learning model for traffic sign detection using capsule networks. arXiv preprint. 2018. arXiv:1805.04424.
85. LaLonde R, Bagci U. Capsules for object segmentation. arXiv preprint. 2018. arXiv:1804.04241.
86. Duarte K, Rawat Y, Shah M. Videocapsulenet: a simplified network for action detection. In: Advances in Neural Information Processing Systems; 2018. pp. 7610–9.
87. Zhao Y, Birdal T, Deng H, Tombari F. 3D point capsule networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2019. pp. 1009–18.
88. Ahmad A, Kakillioglu B, Velipasalar S. 3D capsule networks for object classification from 3D model data. In: 2018 52nd Asilomar Conference on Signals, Systems, and Computers. IEEE; 2018. pp. 2225–9.
89. Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. In: Advances in neural information processing systems; 2015. pp. 2017–25.
90. Worrall DE, Garbin SJ, Turmukhambetov D, Brostow GJ. Harmonic networks: deep translation and rotation equivariance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 5028–37.
91. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer; 2014. p. 818–33.
92. Doersch C, Gupta A, Efros AA. Mid-level visual element discovery as discriminative mode seeking. In: Advances in neural information processing systems; 2013. pp. 494–502.
93. Parizi SN, Vedaldi A, Zisserman A, Felzenszwalb P. Automatic discovery and optimization of parts for image classification. arXiv preprint. 2014. arXiv:1412.6598.
94. Li Y, Liu L, Shen C, Van Den Hengel A. Mining mid-level visual patterns with deep CNN activations. Int J Comput Vision. 2017;121(3):344–64.
95. Yang L, Xie X, Li P, Zhang D, Zhang L. Part-based convolutional neural network for visual recognition. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE; 2017. pp. 1772–6.
96. Kortylewski A, Liu Q, Wang H, Zhang Z, Yuille A. Combining compositional models and deep networks for robust object classification under occlusion. In: The IEEE Winter Conference on Applications of Computer Vision; 2020. pp. 1333–41.
97. Sun Y, Zheng L, Li Y, Yang Y, Tian Q, Wang S. Learning part-based convolutional features for person re-identification. IEEE Trans Pattern Anal Mach Intell. 2019;43(3):902–17. https://doi.org/10.1109/TPAMI.2019.2938523.
98. Hsieh PJ, Lin YL, Chen YH, Hsu W. Egocentric activity recognition by leveraging multiple mid- level representations. In: 2016 IEEE International Conference on Multimedia and Expo (ICME). IEEE; 2016. pp. 1–6.
99. Tang W, Yu P, Wu Y. Deeply learned compositional models for human pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 190–206.
100. Zhang Z, Xie C, Wang J, Xie L, Yuille AL. Deepvoting: a robust and explainable deep network for semantic part detection under partial occlusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 1372–80.
101. Hariharan B, Arbelaez P, Girshick R, Malik J. Object instance segmentation and fine-grained localization using hypercolumns. IEEE Trans Pattern Anal Mach Intell. 2016;39(4):627–39.
102. Johnson J. Deep, skinny neural networks are not universal approximators. arXiv preprint. 2018. arXiv:1810.00393.
103. Marcus G. Deep learning: a critical appraisal. arXiv preprint. 2018. arXiv:1801.00631.
104. Shen X, Tian X, He A, Sun S, Tao D. Transform- invariant convolutional neural networks for image classification and search. In: Proceedings of the 24th ACM international conference on Multimedia; 2016. pp. 1345–54.

105. Shu C, Chen X, Xie Q, Han H. Hierarchical Spatial Transformer Network. arXiv preprint. 2018. arXiv:1801.09467.
106. Wang X, Shrivastava A, Gupta A. A-fast-rcnn: Hard positive generation via adversary for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. pp. 2606–15.
107. Girdhar R, Carreira J, Doersch C, Zisserman A. Video action transformer network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 244–53.
108. Yan X, Yang J, Yumer E, Guo Y, Lee H. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: Proceedings of the 30th International Conference on Neural Information Processing Systems; 2016. pp. 1704–12.
109. Bhagavatula C, Zhu C, Luu K, Savvides M. Faster than real-time facial alignment: a 3D spatial transformer network approach in unconstrained poses. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. pp. 3980–89.
110. Lin CH, Lucey S. Inverse compositional spatial transformer networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 2568–76.
111. Freifeld O, Hauberg S, Batmanghelich K, Fisher JW. Transformations based on continuous piecewise-affine velocity fields. IEEE Trans Pattern Anal Mach Intell. 2017;39(12):2496–509.
112. Wei Z, Sun Y, Lin J, Liu S. Learning adaptive receptive fields for deep image parsing networks. Comput Vis Media. 2018;4(3):231–44.
113. Jing Y, Liu Y, Yang Y, Feng Z, Yu Y, Tao D, et al. Stroke controllable fast style transfer with adaptive receptive fields. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 238–54.
114. Ciresan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012. pp. 3642–9.
115. Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems; 2014. pp. 568–76.
116. Ciresan D, Meier U. Multi-column deep neural networks for offline handwritten Chinese character classification. In: 2015 international joint conference on neural networks (IJCNN). IEEE; 2015. pp. 1–6.
117. Natarajan S, Annamraju AK, Baradkar CS. Traffic sign recognition using weighted multi-convolutional neural network. IET Intel Transport Syst. 2018;12(10):1396–405.
118. Zhang J, Duan S, Wang L, Zou X. Multi-column spatial transformer convolution neural network for traffic sign recognition. In: International Symposium on Neural Networks. Springer; 2018. p. 593–600.
119. Fan C, Li Y, Wang G, Li Y. Learning transformation- invariant representations for image recognition with drop transformation networks. IEEE Access. 2018;6:73357–69.
120. Liu Y, Guo Y, Georgiou T, Lew MS. Fusion that matters: convolutional fusion networks for visual recognition. Multimedia Tools Appl. 2018;77(22):29407–34.
121. Lu X, Lin Z, Shen X, Mech R, Wang JZ. Deep multi- patch aggregation network for image style, aesthetics, and quality estimation. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. pp. 990–8.
122. Wen G, Hou Z, Li H, Li D, Jiang L, Xun E. Ensemble of deep neural networks with probability-based fusion for facial expression recognition. Cogn Comput. 2017;9(5):597–610.
123. Tabik S, Alvear-Sandoval RF, Ruiz MM, Sancho-Gómez JL, Figueiras-Vidal AR, Herrera F. MNIST- NET10: a heterogeneous deep networks fusion based on the degree of certainty to reach 0.1% error rate. Ensembles overview and proposal. Inf Fus. 2020;62:73–80.
124. Hong X, Xiong P, Ji R, Fan H. Deep fusion network for image completion. In: Proceedings of the 27th ACM International Conference on Multimedia; 2019. pp. 2033–42.
125. Gallo I, Calefati A, Nawaz S. Multimodal classification fusion in real-world scenarios. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 5. IEEE; 2017. pp. 36–41.
126. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint. 2014. arXiv:1409.1556.
127. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. pp. 1–9.
128. Xu Y, Xiao T, Zhang J, Yang K, Zhang Z. Scale- invariant convolutional neural networks. arXiv preprint. 2014. arXiv:1411.6369.
129. Liao Z, Carneiro G. Competitive multi-scale convolution. arXiv preprint. 2015. arXiv:1511.05635.
130. Du X, Qu X, He Y, Guo D. Single image super- resolution based on multi-scale competitive convolutional neural network. Sensors. 2018;18(3):789.
131. Chen X, Bin Y, Sang N, Gao C. Scale pyramid network for crowd counting. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE; 2019. pp. 1941–50.
132. Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection. In: Advances in neural information processing systems; 2013. pp. 2553–61.
133. Iandola F, Moskewicz M, Karayev S, Girshick R, Darrell T, Keutzer K. Densenet: implementing efficient convnet descriptor pyramids. arXiv preprint. 2014. arXiv:1404.1869.
134. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint. 2013. arXiv:1312.6229.
135. Wu R, Yan S, Shan Y, Dang Q, Sun G. Deep image: scaling up image recognition. arXiv preprint. 2015;7(8). arXiv:1501.02876.
136. Kong T, Yao A, Chen Y, Sun F. Hypernet: Towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp. 845–53.
137. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. pp. 3431–40.
138. Bell S, Lawrence Zitnick C, Bala K, Girshick R. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp. 2874–83.
139. Cai Z, Fan Q, Feris RS, Vasconcelos N. A unified multi-scale deep convolutional neural network for fast object detection. In: European conference on computer vision. Springer; 2016. p. 354–70.
140. Li Y, Chen Y, Wang N, Zhang Z. Scale-aware trident networks for object detection. In: Proceedings of the IEEE international conference on computer vision; 2019. pp. 6054–63.
141. Zhang Y, Zhou D, Chen S, Gao S, Ma Y. Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. pp. 589–97.
142. Cui J, Chen P, Li R, Liu S, Shen X, Jia J. Fast and practical neural architecture search. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. pp. 6509–18.

143. Cai H, Zhu L, Han S. Proxylessnas: Direct neural architecture search on target task and hardware. arXiv preprint. 2018. arXiv:1812.00332.
144. Cheng G, Han J, Zhou P, Xu D. Learning rotation- invariant and fisher discriminative convolutional neural networks for object detection. IEEE Trans Image Process. 2018;28(1):265–78.
145. Wu F, Hu P, Kong D. Flip-rotate-pooling convolution and split dropout on convolution neural networks for image classification. arXiv preprint. 2015. arXiv:1507.08754.
146. Jiang R, Mei S. Polar coordinate convolutional neural network: from rotation-invariance to translation-invariance. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE; 2019. pp. 355–59.
147. Chen J, Luo Z, Zhang Z, Huang F, Ye Z, Takiguchi T, et al. Polar transformation on image features for orientation-invariant representations. IEEE Trans Multimedia. 2018;21(2):300–13.
148. Kim J, Jung W, Kim H, Lee J. CyCNN: a rotation invariant CNN using polar mapping and cylindrical convolution layers. arXiv preprint. 2020. arXiv:2007.10588.
149. Esteves C, Allen-Blanchette C, Zhou X, Daniilidis K. Polar transformer networks. arXiv preprint. 2017. arXiv:1709.01889.
150. Henriques JF, Vedaldi A. Warped convolutions: efficient invariance to spatial transformations. In: International Conference on Machine Learning. PMLR; 2017. pp. 1461–9.
151. Schmidt U, Roth S. Learning rotation-aware features: from invariant priors to equivariant descriptors. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2012. pp. 2050–7.
152. Amorim M, Bortoloti F, Ciarelli PM, de Oliveira E, de Souza AF. Analysing rotation-invariance of a log-polar transformation in convolutional neural networks. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018. pp. 1–6.
153. Remmelzwaal LA, Mishra AK, Ellis GF. Human eye inspired log-polar pre-processing for neural networks. In: 2020 International SAUPEC/RobMech/PRASA Conference. IEEE; 2020. pp. 1–6.
154. Freeman WT, Adelson EH, et al. The design and use of steerable filters. IEEE Trans Pattern Anal Mach Intell. 1991;13(9):891–906.
155. Cohen TS, Welling M. Steerable CNNs. arXiv preprint. 2016. arXiv:1612.08498.
156. Jacobsen JH, De Brabandere B, Smeulders AW. Dynamic steerable blocks in deep residual networks. arXiv preprint. 2017. arXiv:1706.00598.
157. Weiler M, Hamprecht FA, Storath M. Learning steerable filters for rotation equivariant CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 849–58.
158. Luan S, Chen C, Zhang B, Han J, Liu J. Gabor convolutional networks. IEEE Trans Image Process. 2018;27(9):4357–66.
159. Su YC, Grauman K. Making 360 video watchable in 2d: learning videography for click free viewing. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2017. pp. 1368–76.
160. Monroy R, Lutz S, Chalasani T, Smolic A. Salnet360: saliency maps for omni-directional images with CNN. Signal Process. 2018;69:26–34.
161. Khasanova R, Frossard P. Graph-based isometry invariant representation learning. arXiv preprint. 2017. arXiv:1703.00356.
162. Khasanova R, Frossard P. Graph-based classification of omnidirectional images. In: Proceedings of the IEEE International Conference on Computer Vision Workshops; 2017. pp. 869–78.
163. Cohen TS, Geiger M, Köhler J, Welling M. Spherical CNNs. arXiv preprint. 2018. arXiv:1801.10130.
164. Zhao Q, Zhu C, Dai F, Ma Y, Jin G, Zhang Y. Distortion-aware CNNs for Spherical Images. In: IJCAI; 2018. pp. 1198–204.
165. Zhang Z, Xu Y, Yu J, Gao S. Saliency detection in 360 videos. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 488–503.
166. Perraudin N, Defferrard M, Kacprzak T, Sgier R. DeepSphere: efficient spherical convolutional neural network with HEALPix sampling for cosmological applications. Astronomy Comput. 2019;27:130–46.
167. Boomsma W, Frellsen J. Spherical convolutions and their application in molecular modelling. In: Advances in Neural Information Processing Systems; 2017. pp. 3433–43.
168. Coors B, Paul Condurache A, Geiger A. Spherenet: learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 518–33.
169. Su YC, Grauman K. Learning spherical convolution for fast features from 360 imagery. In: Advances in Neural Information Processing Systems; 2017. pp. 529–39.
170. Esteves C, Allen-Blanchette C, Makadia A, Daniilidis K. Learning so (3) equivariant representations with spherical CNNs. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 52–68.
171. Su YC, Grauman K. Kernel transformer networks for compact spherical convolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 9442–51.
172. Schmalstieg D, Hollerer T. Augmented reality: principles and practice. Addison-Wesley Professional; 2016.
173. Hirabayashi M, Kurosawa K, Yokota R, Imoto D, Hawai Y, Akiba N, et al. Flying object detection system using an omnidirectional camera. Forensic Sci Int. 2020;35:301027.
174. Cohen TS, Geiger M, Weiler M. A general theory of equivariant cnns on homogeneous spaces. In: Advances in Neural Information Processing Systems; 2019. pp. 9145–56.
175. Weiler M, Cesa G. General e (2)-equivariant steerable CNNs. In: Advances in Neural Information Processing Systems; 2019. pp. 14334–45.
176. Kondor R, Trivedi S. On the generalization of equivariance and convolution in neural networks to the action of compact groups. arXiv preprint. 2018. arXiv:1802.03690.
177. Folland GB. A course in abstract harmonic analysis, vol. 29. CRC Press; 2016.
178. Tai KS, Bailis P, Valiant G. Equivariant transformer networks. arXiv preprint. 2019. arXiv:1901.11399.
179. Lenssen JE, Fey M, Libuschewski P. Group equivariant capsule networks. In: Advances in Neural Information Processing Systems; 2018. pp. 8844–53.
180. Romero DW, Bekkers EJ, Tomczak JM, Hoogendoorn M. Attentive group equivariant convolutional networks. arXiv preprint. 2020. arXiv:2002.03830.
181. Worrall D, Welling M. Deep scale-spaces: equivariance over scale. In: Advances in Neural Information Processing Systems; 2019. pp. 7366–78.
182. Marcos D, Kellenberger B, Lobry S, Tuia D. Scale equivariance in CNNs with vector fields. arXiv preprint. 2018. arXiv:1807.11783.
183. Sosnovik I, Szmaja M, Smeulders A. Scale-equivariant steerable networks. arXiv preprint. 2019. arXiv:1910.11093.
184. Romero DW, Bekkers EJ, Tomczak JM, Hoogendoorn M. Wavelet networks: scale equivariant learning from raw waveforms. arXiv preprint. 2020. arXiv:2006.05259.
185. Cheng X, Qiu Q, Calderbank R, Sapiro G. RotDCF: decomposition of convolutional filters for rotation-equivariant deep networks. arXiv preprint. 2018. arXiv:1805.06846.

186. Dieleman S, Willett KW, Dambre J. Rotation- invariant convolutional neural networks for galaxy morphology prediction. Mon Not R Astron Soc. 2015;450(2):1441–59.
187. Cohen TS, weiler M, Kicanaoglu B, Welling M. Gauge equivariant convolutional networks and the icosahedral CNN. In: Proceedings of the 36th International Conference on Machine Learning, 2019:97:1321–30.
188. Worrall D, Brostow G. Cubenet: equivariance to 3D rotation and translation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. pp. 567–84.
189. Cohen TS, Welling M. Transformation properties of learned visual representations. arXiv preprint. 2014. arXiv:1412.7659.
190. Smets B, Portegies J, Bekkers E, Duits R. PDE-based group equivariant convolutional neural networks. arXiv preprint. 2020. arXiv:2001.09046.
191. Romero DW, Hoogendoorn M. Co-attentive equivariant neural networks: Focusing equivariance on transformations co-occurring in data. arXiv preprint. 2019. arXiv:1911.07849.
192. Romero DW, Cordonnier JB. Group equivariant stand-alone self-attention for vision. arXiv preprint. 2020. arXiv:2010.00977.
193. Finzi M, Stanton S, Izmailov P, Wilson AG. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. arXiv preprint. 2020. arXiv:2002.12880.
194. Bruna J, Mallat S. Invariant scattering convolution networks. IEEE Trans Pattern Anal Mach Intell. 2013;35(8):1872–86.
195. Bekkers EJ. B-spline CNNs on lie groups. arXiv preprint. 2019. arXiv:1909.12057.
196. Fey M, Eric Lenssen J, Weichert F, Mu¨ller H. Splinecnn: fast geometric deep learning with continuous b-spline kernels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 869–77.
197. Itti L, Koch C. Computational modelling of visual attention. Nat Rev Neurosci. 2001;2(3):194–203.
198. Dey N, Chen A, Ghafurian S. Group equivariant generative adversarial networks. arXiv preprint. 2020. arXiv:2005.01683.
199. Shen C, Wang X, Song J, Sun L, Song M. Amalgamating knowledge towards comprehensive classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. pp. 3068–75.
200. Carlucci FM, D'Innocente A, Bucci S, Caputo B, Tommasi T. Domain generalization by solving jigsaw puzzles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 2229–38.
201. Finn C, Abbeel P, Levine S. Model-agnostic meta- learning for fast adaptation of deep networks. arXiv preprint. 2017. arXiv:1703.03400.
202. Jarvers C, Neumann H. Incorporating feedback in convolutional neural networks. In: Proceedings of the Cognitive Computational Neuroscience Conference; 2019. pp. 395–8.
203. Marblestone AH, Wayne G, Kording KP. Toward an integration of deep learning and neuroscience. Front Comput Neurosci. 2016;10:94.
204. Hu T, Yang P, Zhang C, Yu G, Mu Y, Snoek CG. Attention-based multi-context guiding for few-shot semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. pp 8441–8.
205. Hutter F, Kotthoff L, Vanschoren J. Automated machine learning: methods, systems, challenges. Springer Nature; 2019.
206. He X, Zhao K, Chu X. AutoML: a survey of the state-of-the-art. arXiv preprint. 2019. arXiv:1908.00709.
207. Zoph B, Le QV. Neural architecture search with reinforcement learning. arXiv preprint. 2016. arXiv:1611.01578.
208. Peng J, Sun M, ZHANG ZX, Tan T, Yan J. Efficient neural architecture transformation search in channel- level for object detection. In: Advances in Neural Information Processing Systems; 2019. pp. 14313–22.
209. Nekrasov V, Chen H, Shen C, Reid I. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In: Proceedings of the IEEE Conference on computer vision and pattern recognition; 2019. pp. 9126–35.
210. Zhang Y, Qiu Z, Liu J, Yao T, Liu D, Mei T. Customizable architecture search for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 11641–50.
211. Liu C, Chen LC, Schroff F, Adam H, Hua W, Yuille AL, et al. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2019. pp. 82–92.
212. Elsken T, Staffler B, Metzen JH, Hutter F. Meta-learning of neural architectures for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. pp. 12365–75.
213. Biedenkapp A, Bozkurt HF, Eimer T, Hutter F, Lindauer M. Dynamic algorithm configuration: foundation of a new meta-algorithmic framework. In: Proceedings of the Twenty-fourth European Conference on Artificial Intelligence (ECAI'20) (Jun 2020); 2020.
214. Elsken T, Metzen JH, Hutter F. Simple and efficient architecture search for convolutional neural networks. arXiv preprint. 2017. arXiv:1711.04528.
215. Veniat T, Denoyer L. Learning time/memory-efficient deep architectures with budgeted super networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. pp. 3492–500.
216. Jin H, Song Q, Hu X. Auto-keras: An efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019. pp. 1946–56.
217. Feurer M, Klein A, Eggensperger K, Springenberg JT, Blum M, Hutter F. Auto-sklearn: efficient and robust automated machine learning. In: Automated machine learning. Cham: Springer; 2019. p. 113–34.