AEGAEUM JOURNAL

Computer vision In Artificial Intelligence

ISSN NO: 0776-3808

Gauresh Rajendra Badgujar Dr.Mrs.Pratibha Adkar

MCA Department , PES Modern College Of Engineering

Abstract:

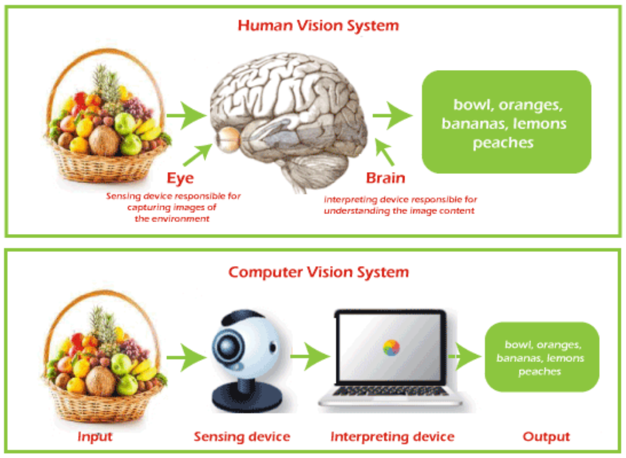
The rapid advancements in computer vision, a key area within artificial intelligence, have been primarily driven by innovations in deep learning and neural networks. This paper surveys recent developments in computer vision techniques and their wide-ranging applications. We begin by examining fundamental computer vision algorithms, covering both traditional methods such as feature extraction and modern approaches like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The superior capabilities of deep learning models in processing intricate visual data are emphasized, highlighting their success in tasks such as object detection, image classification, semantic segmentation, and instance segmentation. Additionally, the integration of computer vision with other AI domains is explored, encompassing robotics, autonomous vehicles, healthcare, and surveillance. In robotics, computer vision is crucial for enabling autonomous perception and interaction with the environment. For autonomous vehicles, vision-based systems are vital for navigation, object detection, and scene understanding. In healthcare, computer vision facilitates medical image analysis, disease diagnosis, and treatment planning. Moreover, vision algorithms are integral to surveillance and security applications, aiding in facial recognition, anomaly detection, and activity recognition.

Keywords: Computer Vision, Artificial Intelligence, Deep Learning, Neural Networks, Image Processing

Introduction:

The field of computer vision, which enables machines to interpret and understand visual information from the world, has seen a dramatic evolution, becoming fundamental to contemporary AI systems. Unlike the human visual system, which effortlessly identifies objects and scenes, replicating this capability in machines has been a significant objective of AI research. Earlier computer vision algorithms depended on handcrafted features and shallow models for tasks such as object detection and image classification. However, the advent of deep learning, especially convolutional neural networks (CNNs), has transformed the field by providing exceptional visual perception and analytical abilities.

Deep learning models, leveraging vast datasets and robust computational resources, have achieved, and at times surpassed, human-level performance in visual recognition tasks. CNNs, with their capability to autonomously learn hierarchical features from raw pixel data, have driven advancements across a variety of applications, including autonomous vehicles, robotics, healthcare, and surveillance.


Human Vision System
bowl, oranges, bananas, lemons peaches
Eye
Sensing device responsible for capturing images of the environment
Brain
interpreting device responsible for understanding the image content
Computer Vision System
bowl, oranges, bananas, lemons peaches
Input
Sensing device
Interpreting device
Output

Literature Survey:

"Computer Vision: Algorithms and Applications" by Richard Szeliski: This extensive book provides a thorough examination of computer vision algorithms, including topics like image formation, feature detection, image segmentation, and object recognition. It is a valuable foundational resource for those seeking to grasp the core principles and methodologies of computer vision.

"Deep Learning for Computer Vision" by Rajalingappaa Shanmugamani: This book zeroes in on deep learning methods for computer vision, detailing convolutional neural networks (CNNs), recurrent neural networks (RNNs), and their uses in image classification, object detection, and image generation. It features practical examples and code implementations to facilitate hands-on learning.

"Computer Vision: Models, Learning, and Inference" by Simon J.D. Prince: Offering a thorough introduction to computer vision, this textbook covers image formation, camera models, feature extraction, and probabilistic models for object recognition. It highlights a unified approach to addressing vision problems through probabilistic inference.

"Handbook of Computer Vision Algorithms in Image Algebra" edited by Gerhard X. Ritter and Joseph N. Wilson: This handbook compiles a variety of algorithms and techniques for different computer vision tasks, structured within the context of image algebra. Topics include image filtering, edge detection, image morphological operations, and shape analysis, providing both theoretical underpinnings and practical implementations.

"Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce: This textbook offers a contemporary view of computer vision, blending classical techniques with modern deep learning approaches. It covers image formation, feature detection, image segmentation, and object recognition, giving insights into both traditional methods and the latest advancements in the field.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are at the forefront of performance in numerous computer vision tasks, such as image classification, object detection , and image parsing . Recently, significant attention has been directed towards improving CNN components, including pooling layers, activation functions , and nonlinear layers . These improvements either facilitate better CNN training or enhance the network's learning capabilities. Our approach, however, boosts CNN performance from a different perspective by designing a novel hierarchical architecture that integrates an existing CNN model as a foundational element. We embed multiple foundational CNN units into a larger, more complex hierarchical deep CNN framework.

Recurrent Neural Networks (RNNs) for Sequence Modeling

Recurrent Neural Networks (RNNs) are adept at handling sequences of variable lengths. A notable variant, Long Short-Term Memory (LSTM) networks , has demonstrated success in natural language processing tasks such as machine translation.

Combining RNNs and CNNs for Visual Content Description

The combination of RNNs and CNNs has proven effective in generating descriptions for visual content, including still images and videos . This is achieved by employing an RNN model that first processes the visual content and then predicts a sequence of words describing it. Our work expands on this concept by developing a model for question answering, where the model generates responses based on both visual and natural language inputs.

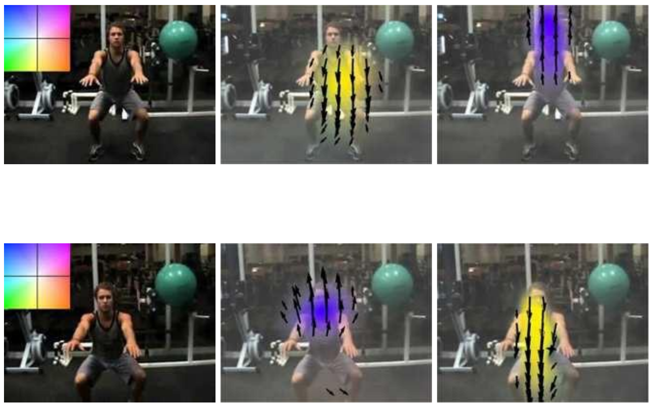
Implementation Details

In this section, we outline the specifics of our model's training process. We implemented our network architecture using the Caffe library . During training, the weights in all convolutional layers are kept constant. We train the fully connected layers to optimize the least squares error of the regression criterion outlined previously. To ensure regularization, we apply a dropout rate of 0.5 in the fully connected layers.

Our training dataset comprises 80,000 samples from the BSDS500 dataset . As previously mentioned, the labels represent the consensus among human annotators regarding boundary presence. We divide the label space into four quartiles and select an equal number of samples from each quartile to balance the dataset. Additionally, we use a hold-out dataset of 40,000 samples for hard-positive mining to reduce false-negative predictions. For the initial 25 epochs, we train the network on the original 80,000 samples. Subsequently, we test the network on the hold-out dataset to identify false negatives. We then augment the training set with these false negatives and an equal number of randomly selected true negatives, continuing training for an additional 25 epochs on this expanded dataset.

Methods

Our objective is to learn a mapping from input RGB images to an output space representing the predicted motion of each pixel in terms of optical flow. We propose using CNNs for this task. However, several key questions need addressing: What constitutes an effective output space? What is an appropriate loss function? Should optical flow prediction be framed as a regression or a classification problem? What architecture best suits this problem? Below, we delve into these considerations.



(a) Input Image (b) Prediction (c) Ground Truth

Fig 3

Figure 3 Analysis

Consider the images on the left: is the man squatting up or down? The bottom image depicts the near completion (or beginning) of the action, while the top image captures the motion mid-way. Our dataset includes many such ambiguous images. In our evaluation, we consider the distribution of movements predicted by our network. While it is highly probable that the man will move up or down, it is unlikely he will veer left or right.

Regression as Classification

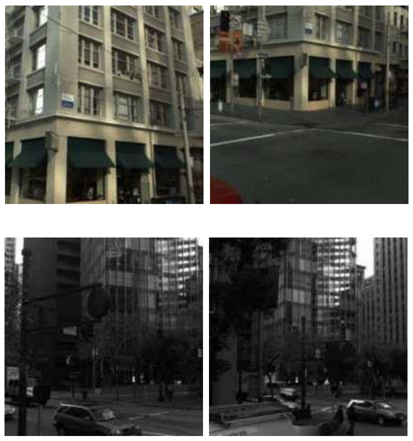
Motion estimation intuitively aligns with a regression approach due to its continuous nature. This methodology was employed in [20], where structured random forests were used to regress the magnitude and direction of optical flow. However, this approach tends to smoothen results as ambiguity is averaged out. In surface normal prediction, a similar challenge is addressed by reframing structured regression as a classification problem [21, 22]. By quantizing surface normal vectors into clusters, the problem becomes one of predicting cluster membership.

In our work, we adopt a similar strategy for optical flow vectors, quantizing them into 40 clusters using k-means. This allows us to treat the task like semantic segmentation, classifying each image region into a specific optical flow cluster. We utilize a soft-max loss layer for gradient computation, but at test time, we generate a soft output by calculating a weighted-probability sum over all clusters for each pixel. This transformation into classification yields a discrete probability distribution over vector directions and magnitudes.

Given the inherent ambiguity in motion prediction (see Figure 3), we can use this probability distribution to assess the informativeness of our predictions. For instance, while we may not be certain if the man in Figure 3 is sitting down or standing up, we can be confident he will not turn left or right. Consequently, our network can rank upward and downward facing clusters higher than other directions. Even if the highest-ranked cluster contradicts the ground truth, the next highest-ranked cluster might still be correct. A discrete probability distribution provides a clearer understanding of our network’s performance. Additionally, calculating the distribution's entropy allows us to gauge the confidence in our motion predictions and identify images with higher likelihoods of accuracy.

A Simple Model of Motion-Based Learning

We simulate the agent's visual system using a Convolutional Neural Network (CNN) [23]. The agent refines its visual representations by minimizing the error between the egomotion information (camera transformation) derived from its motor system and the egomotion predicted using visual inputs alone. This task is akin to training a Siamese Style CNN (SCNN) [24] with two input images, predicting the egomotion experienced by the agent as it moved between the two spatial points where the images were captured. To learn effective visual representations, the agent continuously performs this task. task as it moves around in its environment. In this work we use the pretraining-finetuning paradigm for evaluating the utility of learnt features. Pretraining is the process of optimizing the weights of a randomly initialized CNN for an auxiliary task that is not the same as the target task. Finetuning is the process of modifying the weights of-



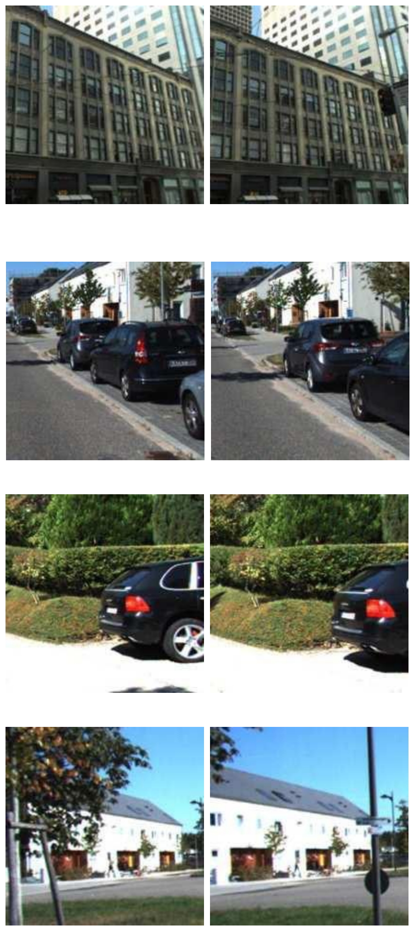


Figure 1: Utilizing Egomotion for Visual Feature Learning

This figure illustrates the application of egomotion as supervision for learning valuable visual features. A mobile agent equipped with visual sensors captures a sequence of images as it navigates its environment. The agent's movement corresponds to the movement of a camera. In this study, egomotion-based learning is framed as the task of predicting camera transformation from pairs of images. The top and bottom rows of the figure display sample image pairs from the SF and KITTI datasets, respectively, utilized for feature learning.

Two-Stream Architecture

The CNN architecture comprises two streams, each independently computing features for one image. Both streams share the same architecture and weights, performing identical operations for feature computation. These individual streams are referred to as BaseCNNs (BCNNs). The features computed by two BCNNs are concatenated and forwarded to another CNN, termed TopCNN (TCNN), as illustrated in Figure 2. TCNN is responsible for utilizing the BCNN features to predict the camera transformation between the input image pair. Following pretraining, TCNN is removed, and a single BCNN is employed as a standard CNN for feature computation in the target task.

CNN Architecture Notation

In our notation, abbreviations like Ck, Fk, P, D, and Op represent a convolutional (C) layer with k filters, a fully-connected (F) layer with k filters, pooling (P), dropout (D), and the output (Op) layers, respectively. We incorporated ReLU non-linearity after every convolutional/fully-connected layer, excluding the output layer. A dropout layer with a dropout rate of 0.5 was consistently employed. The output layer consisted of a fully connected layer with units equal to the desired number of outputs. For instance, C96-P-F500-D denotes a network with 96 filters in the convolutional layer, followed by ReLU non-linearity, a pooling layer, a fully-connected layer with 500 units, ReLU non-linearity, and a dropout layer.

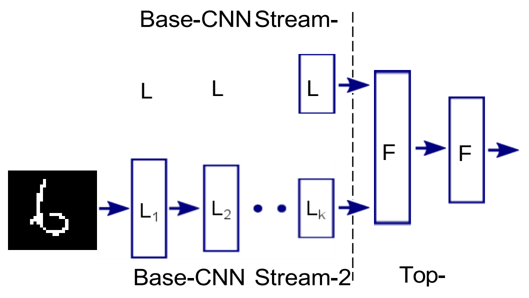


Figure 2: Method Description for Feature Learning
The method for feature learning is depicted in this figure. Visual features are acquired by training a Siamese-style Convolutional Neural Network (SCNN) [8], which takes two images as inputs and predicts the transformation between them (i.e., egomotion). Each stream of the SCNN, referred to as Base-CNN or BCNN, computes features for one image. The outputs of two BCNNs are concatenated and fed as inputs to a second multilayer CNN known as TopCNN (TCNN) (illustrated as layers F1 and F2). The two BCNNs share the same architecture and weights. Following feature learning, TCNN is discarded, and a single BCNN stream is utilized as a standard CNN for extracting features for target tasks like scene recognition.

Slow Feature Analysis (SFA) Baseline

Slow Feature Analysis (SFA) is a feature learning method based on the notion that useful features exhibit slow changes over time. We employed a contrastive loss formulation of SFA [23, 25], defined as:

$$L(x_{t1}, x_{t2}, W) = \begin{cases} D(x_{t1}, x_{t2}) & \text{if} \ |t1 - t2| \leq T \\ 1 - \max(0, m - D(x_{t1}, x_{t2})) & \text{if} \ |t1 - t2| > T \end{cases}$$

Here, L is the loss, x_{t1} and x_{t2} are feature representations of frames observed at times $t1$ and $t2$, W are the parameters specifying the feature extraction process, D is a distance measure with parameter m as a predefined margin, and T is a predefined time threshold determining temporal closeness between frames.

Proof of Concept using MNIST
On the MNIST dataset, egomotion was simulated by generating synthetic data comprising random transformations (translations and rotations) of digit images. CNNs were trained to predict transformations between these image pairs. Egomotion-based pretraining involved constraining relative translations and rotations within specific ranges and posing transformation prediction as a classification task. For SFA-based pretraining, image pairs with relative translations and rotations within certain ranges were considered temporally close.

Network Architectures

Multiple BCNN architectures were experimented with, and the optimal architecture for each pretraining method was selected separately. In egomotion-based pretraining, the two BCNN streams were concatenated using the TCNN architecture: F1000-D-Op.
[26] was utilized for training all models.

Table 1: Comparison of Pretraining Methods on MNIST

The comparison table demonstrates that egomotion-based pretraining... [content from Table 1 goes here] outperforms many previous approaches for unsupervised learning. The performance is reported as the error rate.

Method	# examples for finetuning			
	100	300	1000	10000
Autoencoder [17]	24.1	12.2	7.7	4.8
Ranzato et al. [29]	-	7.18	3.21	0.85
Lee et al. [22]	-	-	2.62	-
Train from Scratch	20.1	8.3	4.5	1.6
SFA (m=10)	11.2	6.4	3.5	2.1
SFA (m=100)	11.9	6.4	4.8	4.7
Egomotion (ours)	8.7	3.6	2.0	0.9

Pretraining involved 40,000 iterations (equivalent to 5 million examples) with an initial learning rate of 0.01, halved every 10,000 iterations. During fine-tuning for digit classification, the BCNN architecture utilized was BCNN-F500-D-Op. To assess BCNN feature quality, all BCNN layer learning rates were set to 0 during fine-tuning. Fine-tuning lasted 4,000 iterations (equivalent to 50 epochs for 10,000 labeled training examples) with a fixed learning rate of 0.01.

Results

BCNN features were evaluated by computing error rates for digit classification using 100, 300, 1,000, and 10,000 class-labeled training examples, randomly sampled from the standard training set of 60,000 digits. Original digit images (without transformations or data augmentation) were used. The standard test set of 10,000 digits was employed for evaluation, and error rates averaged across 3 runs are presented in Table 1. The BCNN architecture C96-P-C256-P was found optimal for egomotion and SFA-based pretraining and for training from scratch (i.e., random weight initialization). Supplementary material provides results for other architectures. SFA-based pretraining experimented with various margin values and found that m = 10, 100 resulted in the best performance. Our method outperformed convolutional deep belief networks [27] and a previous approach based on learning features invariant to transformations [28] and SFA-based pretraining.

Conclusion

This paper has surveyed significant advancements in computer vision techniques and their applications across diverse domains. Integration of deep learning models, particularly CNNs and RNNs, has transformed the field, enabling tasks like object detection, image classification, and scene understanding with high accuracy. The intersection of computer vision with robotics, autonomous vehicles, healthcare, and surveillance underscores its critical role in modern AI applications. Future research is poised to further explore and enhance these techniques, broadening their capabilities and applications.

REFERENCE

[1]A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS,2012.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014.

[4] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.

[5] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In ICML, 2013.

[6]J. T. Springenberg and M. Riedmiller. Improving deep neural networks with probabilistic maxout units. arXiv preprintarXiv:1312.6116, 2013.

[7] M. Lin, Q. Chen, and S. Yan. Network in network. CoRR,2013.

[8] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. In ICLR,2013.
[9] S. Hochreiter and J. Schmidhuber. Long short-term memory.
Neural Computation, 1997.
[10] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares,
H. Schwenk, D. Bahdanau, and Y. Bengio. Learning phrase
representations using rnn encoder-decoder for statistical ma-
chine translation. In EMNLP, 2014.
[11] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach,
S. Venugopalan, K. Saenko, and T. Darrell. Long-term recur-
rent convolutional networks for visual recognition and de-
scription. In CVPR, 2015.
[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In CVPR, 2015.
[13] I. Sutskever, O. Vinyals, and Q. V. V. Le. Sequence to se-
quence learning with neural networks. In NIPS. 2014.
[14] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach,
R. Mooney, and K. Saenko. Translating videos to natural
language using deep recurrent neural networks. In NAACL,
2015.
[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and
tell: A neural image caption generator. arXiv:1411.4555,
2014
[16] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the
visual interpretation of sentences. In ICCV, 2013.
[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
[18] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In ICCV, 2011
[19] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proc. 8th Int'l Conf. Computer Vision, volume 2, pages 416–423, July 2001.

[20] S. L. Pintea, J. C. van Gemert, and A. W. Smeulders. Dej´ a` vu: Motion prediction in static images. In ECCV. 2014. 2, 3, 4, 5, 6, 7

[21] B. Z. Lubor Ladicky and M. Pollefeys. Discriminatively ` trained dense surface normal estimation. 3

[22] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. arXiv preprint arXiv:1411.4958,

[23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989

[24] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society.

[25] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In

Proceedings of the 26th Annual International Conference on Machine Learning, pages 737–744. ACM, 2009

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of

[27] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 609–616. ACM, 2009.

[28] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.