Deep Learning Projects for Multidisciplinary Engineering Design Students

Mr. Robert L. Avanzato, Pennsylvania State University, Abington

Robert Avanzato is an associate professor of engineering at the Penn State Abington campus where he teaches courses in electrical and computer engineering, computer science, and robotics. His research interests are mobile robotics, artificial intelligence, computer vision, deep learning and virtual environments.

Deep Learning Projects for Multidisciplinary Engineering Design Students

Abstract

Deep Learning is a form of AI machine learning that has gained a great deal of recognition in the past 10 years in a wide range of areas such as medical diagnosis, quality assurance, defect detection, face detection, autonomous vehicles, and many others. Deep learning networks, however, typically require large training databases of labeled images and often require specialized hardware and high-level software expertise. Techniques, such as transfer learning and the proper choice of software tools can mitigate some of these requirements. This paper describes a new, project-based course module to introduce deep learning and computer vision to undergraduate multidisciplinary engineering students in a robotics design and applications course using MATLAB software.

1. Introduction

Deep Learning (DL) is a form of AI machine learning that has gained a great deal of recognition in the past 10 years due to numerous success stories in areas such as computer vision, language translation, face detection, medical diagnosis and treatment, visual inspection, manufacturing, etc. [1-6] Deep learning networks can often outperform human experts in many areas, such as identifying tumors in medical images or identifying defects in the manufacturing process. Deep learning is described as "end to end" learning because it encompasses low-level feature extraction and classification algorithms. This contrasts to traditional machine learning using neural networks which require a manual feature extraction step to be performed by engineers before training a neural network. Deep learning models, in the area of computer vision (CV) applications, are most often implemented as a convolutional neural network (CNN) and consist of a large number of hidden layers (10 to 100 or more). In very basic terms, each layer of the network detects features at a higher level as the processing moves from input image to the output. For example, the first layers of a CNN may learn to detect basic features such as edges and corners etc., and more advanced layers may detect higher level features such as facial features and traffic signs. These networks, however, typically require large training databases of labeled images (10K to 1M images or more) and often require specialized graphical processor units (GPU) hardware support and specialized software expertise to train the networks. However, techniques such as "transfer learning" [7,8] can often be employed to mitigate some of these requirements. Also, the judicious use of software tools, such as MATLAB, can soften the learning curve for students to participate in deep learning projects.

The goal of this project was to develop a project-based educational module to introduce multidisciplinary engineering design undergraduates to deep learning projects in a senior robotics applications and design course. The course already covers a module on traditional computer vision and image processing algorithms including color detection, open/close functions, noise removal, edge detection, blob analysis, Hough transforms, filtering (correlation

and convolution). In the new module, deep learning is introduced using MATLAB Deep Learning Toolbox. Most importantly, "transfer learning" is utilized to minimize training time and eliminate the need for GPU support. Using transfer learning, students can add customized layers to previously training networks (such as Alexnet and GoogleNet) and thereby, avoid the need for GPU support and allow the use of a smaller database of labeled images (30 to 100 images) for training. Transfer learning leverages the notion that the lower layers of the existing trained network (such as Alexnet) will extract features that are relevant to a new task, and so the entire network with the new objects to be detected need not be retrained.

The purpose of this paper is to present an outline of the curriculum, examples of assigned student projects and educational strategies for the Deep Learning module and to discuss the successful project ideas completed by the students in the course over the past 2 years. Deep learning and computer vision have been generally under the purview of computer science and data science majors, but software tools such as MATLAB and carefully selected curriculum can make AI and DL accessible to multidisciplinary engineering students who generally possess limited and varied software development backgrounds.

It should be noted that an application of deep learning models called large language models (LLMs) have been introduced into the mainstream in the past few years. Examples include generative AI models such as the text-to-text model, chatGPT [9], GPT-4 [9], and others. LLMs use a transformer model architecture instead of a CNN and the transformer architectures are currently being explored for use in computer vision applications. Models such as chatGPT (and others) have also proven useful for programming code generation and productivity enhancement. These LLMs are of growing importance but are outside the scope of this project and paper.

2. Deep Learning Curriculum and Projects

The specific goal of this project is to design and implement an instructional 7-week course module to introduce deep learning and computer vision with a project-based orientation. As mentioned, the target course is a senior-level engineering design course in robotics applications required for a B.S. Engineering degree with a Multidisciplinary Engineering Design option offered through Penn State Abington and Brandywine campuses. The first 7 weeks of the course focuses on image processing and traditional computer vision (non-AI). For this new deep learning course module, the learning outcomes developed would enable a student to: 1) communicate the role of DL and CNNs in computer vision tasks in robotics and industry, 2) implement and test DL algorithms in the areas of image classification and object detection, 3) design, integrate and implement a DL solution into a mobile robotics and/or robot manipulator application. One of the main challenges of this module was to select software development platform that matches and supports the academic background of the students, who are majoring in a multidisciplinary engineering design option. This program and design option blends mechanical, electrical, computer, design and systems engineering components. Students entering this course have a varied background in computer software which includes a single course in one of several languages, including, C++, Python and MATLAB. It was imperative from the start

that the module in Deep Learning and CV would not be the same as a module taught to computer science majors or data scientists who have a much deeper familiarity with programming languages. For this reason, MATLAB was chosen as the software package for the DL module. For one, students were familiar with MATLAB from a previous robotics course, and the earlier course module on image processing. MATLAB also presents a more shallow learning curve, is interactive, and supports prototyping and visualization. MATLAB fully supports CV and DL with the Computer Vision Toolbox and Deep Learning Toolbox. Another constraint was the required use of low-cost hardware and limited laboratory resources. As mentioned, all the network training was used with standard laptops with CPUs and minimal GPU support. No specialized GPU hardware was required.

Student projects focused on computer vision applications in robotics and manufacturing such as visual defect analysis involved identifying good/broken cookies on a conveyor belt, missing or misaligned bottle caps on small bottles, manufacturing part identification (screws, bolts, etc.), identifying food products, and others. An undergraduate honors thesis was completed by a student using this course deep learning curriculum to identify surface defects on 3D printed objects on a conveyor belt which were then removed by a low-cost robot arm.

There are 2 key areas of deep learning as applied to computer vision – one key area is image classification, and the other key area is object detection. In image classification, for example, a single input image can be identified as an image containing a defect or object, but the exact location and size of the defect or object in the image is not determined. In object detection, a bounding box surrounding one of more detected defects or objects of interest in an image would be determined. Each method has advantages and disadvantages, and each method has a different strategy for data collection, modelling, and training. Both of these approaches are considered "supervised" learning since the models are trained on labeled data. The student projects described in the following section represent both image classification and object detection (using the Yolo algorithm [10]). Other related deep learning technologies such as segmentation and reinforcement learning are outside the objectives the course goals.

Presented below is the course module curriculum on a weekly basis for a total of 7 weeks (the full semester is 15 weeks). The class meets twice per week and each class meeting is 75-minutes. The students have access to a lab to perform project work outside of the scheduled class times.

Week #	Topics (with key HW, Labs, Projects)
1A	- AI and Deep Learning (DL) - What is Deep Learning? - Alexnet and MATLAB - Examples of DL and Machine Learning (ML) in industry - DL versus traditional computer vision (CV) - **HW: Research industrial or R&D application of DL** - **HW: Test Alexnet image classification network using MATLAB and webcam**
1B	- Applications of DL in manufacturing, healthcare, fraud detection, etc. - Computer vision applications

	-	Convolutional Neural Network (CNN) Architecture (conceptual)
	-	Convolutional kernel (basis for CNN)
	-	Potential Network Training Problems
	-	Human Bias and Ethics
	-	DL applied to photography (examples)
	-	NMIST database of digit images
2	-	Transfer Learning
	-	MATLAB code for transfer learning; hyperparameters
	-	Advantages of transfer learning
	-	Overfitting in DL networks
	-	Overview of pretrained networks
	-	Interfacing MATLAB to other DL networks (Tensorflow, PyTorch, etc.)
	-	**Project: DL image classifier for cracker/cookie defect detector using MATLAB**
3	-	Data augmentation
	-	Intro to object detection (Yolo algorithm)
	-	Labeling ground truth for object detection
	-	Kaggle dataset repository
4	-	Discussion of gradient descent
	-	Overfitting and underfitting
	-	Epochs, mini batch size, iterations
	-	Accuracy vs Epochs plots
	-	MATLAB code for image classification
	-	Confusion chart
	-	**Project: Parametric study for image classification (bottle lid defect detection or alternative) using MATLAB**
5	-	Image classification versus object detection
	-	YOLO algorithm basic features and architecture
	-	Using MATLAB imagelabeler tool for ground truth labeling
	-	MATLAB code for Yolo v2
	-	**Project: Orange cone object detection using YOLO algorithm and MATLAB**
6 - 7	-	**Team project: Integrate DL solution into mobile robot or robot arm system using MATLAB. Use either Arduino robot with ESP32-CAM, Turtlebot 3 (ROS) or Clearpath Husky (ROS) mobile robot platforms or Dobot robot arm technology (ROS)**
	-	Test results, document design and integration, demonstrate operation, present results

3. Examples of Deep Learning Projects in the Curriculum

The first project example to be discussed is a bottle cap detection system. In this project, a deep learning system has been developed to determine if a bottle cap on a small plastic bottle is a) installed properly, b) misaligned, c) open, or d) missing. Examples for each case or category are displayed in the table below (figure 1). The task for the students is to develop and test an image classification CNN using transfer learning and MATLAB to distinguish and identify the various bottle cap orientations. A training set can be developed by the student teams, or they can utilize an existing bottle image dataset developed by a previous class (30 images per category). The use of existing image datasets can reduce the time required to complete laboratories. Other objects can be optionally chosen by the students in the general area of defect detection and quality

inspection.   Each set of 30 images is stored in separate folder and MATLAB tools handle the dataset data structures and preprocessing (resizing, etc.) of the images. The students download an existing pre-trained deep learning network, such as Alexnet, make minor modifications and set training hyperparameters such as number of epochs, minibatch size and learning rates. Alexnet is a deep learning network that was trained with over a million images [11].

Lid Open	Lid Misaligned	Lid Missing	Lid Properly Placed
30 images	30 images	30 images	30 images

Figure 1: Training dataset sample images for bottle lid detection system

In this transfer learning technique, however, only the final layers of the Alexnet network (or any other large, pretrained network) are trained with the images collected by the students. This approach reduces the need to train an entire network which would be costly in terms of computation and speed. A parametric study was performed in which # of epochs, minibatch size and learning rate were modified and the accuracy was reported in each case. Real world testing was performed with a standard webcam and also an ESP32-CAM wifi camera. Due in part to the lack of complexity in the image background, the accuracy is typically above 90% and training converges within approximately 5 – 10 epochs, requiring less than a few minutes of CPU time on a typical laptop computer. MATLAB generates analytics during network training which are used by the students to evaluate the models. The engineering context for this project is quality control and defect detection in a production or manufacturing environment. This project can be potentially integrated with a simple robot arm and conveyor belt to expose students to an intelligent manufacturing design application. Due to the MATLAB tool, this project can be completed individually or in teams of 2 students in a period of 1 to 1.5 weeks.

In the second project to be discussed in this paper, students developed an object detection, deep learning solution to identify small orange (or red) traffic cones using the Yolo object detection algorithm in MATLAB. The engineering application or context is to aid in the navigation of a mobile robot. The first step is to collect camera images of orange cones and use a MATLAB tool to label the images. In this step the user must manually draw a bounding box around the orange cones located in a set of images captured by the students. Optionally, training could occur

with other objects such as traffic signs, hand gestures etc. The figure below (figure 2) is a
screenshot of one of the image labeler tools in MATLAB. It is also possible to label objects in
videos as well using other MATLAB-supported tools. The dataset size for this project was 20 to
30 images. Once the images are labeled, the image data with bounding box information is
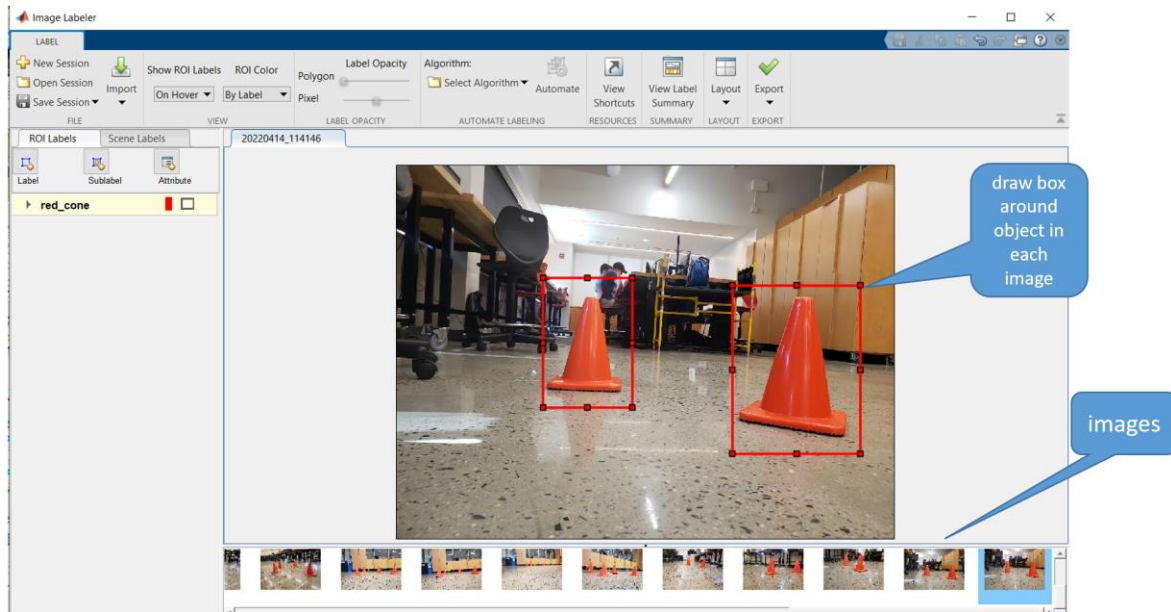exported directly into the MATLAB environment for processing with the Yolo algorithm.

Figure 2: Groundtruth (image labeling) tool using MATLAB

After the object detection is trained, the students test the network with a web camera and test
with real world data. As can be seen below (figure 3), the algorithm has detected two orange
(red) cones and is able to draw a bounding box around each detected cone. The bounding boxes
determine the location of the orange cones in the image and this data is useful when this network
is integrated into a mobile robot project to support navigations. As with the previous project this
lab can be completed in a 1- to 1.5- week period individually or with a small team of 2 students
within the 7-week module.

Figure 3: Traffic cone detection with Yolo object detection algorithm

In the final project it was required to integrate a DL network into a robot platform solution such as a mobile robot or a robot arm. In the past two years we have been focusing on small mobile robots such as an Arduino-based, differential-drive, mobile robot with an ESP32-CAM camera. The students also have option of integration with a Turtlebot 3 (ROS enabled mobile robot with camera from Robotis.com) or a Husky mobile robot platform (clearpathrobotics.com). Due to the fact that both the Turtlebot 3 and Husky robot can be operated using ROS in a MATLAB environment, the integration of robot control with computer vision and deep learning, also implemented in MATLAB, is simplified. Our small desktop robot arms (Dobot) and conveyor belts are also controlled by ROS in the MATLAB environment on a Windows machine and have been used for deep learning computer vision projects. This development and implementation strategy is very useful in an educational environment that possesses limited resources. This also allows educators to focus more time on design concepts and engage students in the integration and implementation of functional systems and spend less time with complex code development.

Below is a block diagram of a simple system incorporating a low-cost Arduino robot (Osoyoo), an ESP32-CAM wifi camera, and a Windows laptop running MATLAB with a deep learning network (see figure 4). MATLAB is first used to train a network such as an object detection algorithm to detect orange traffic cones as described above. In the implementation phase, the camera feed from the robot is analyzed by the trained network and the detected orange cone locations are used to direct the robot motion (steering and speed) by way of Bluetooth (BT) commands sent to the robot. A very basic lab would be to navigate between or along orange cones to reach a destination. This is also a lab which can be undertaken in a short period (1.5 to 2 weeks) by students using low-cost hardware and traditional laptops running MATLAB. Although the video frame rates are generally slow in this configuration (5 to 10 fps) the goal of systems integration of an AI deep learning system with a mobile robot is achieved.
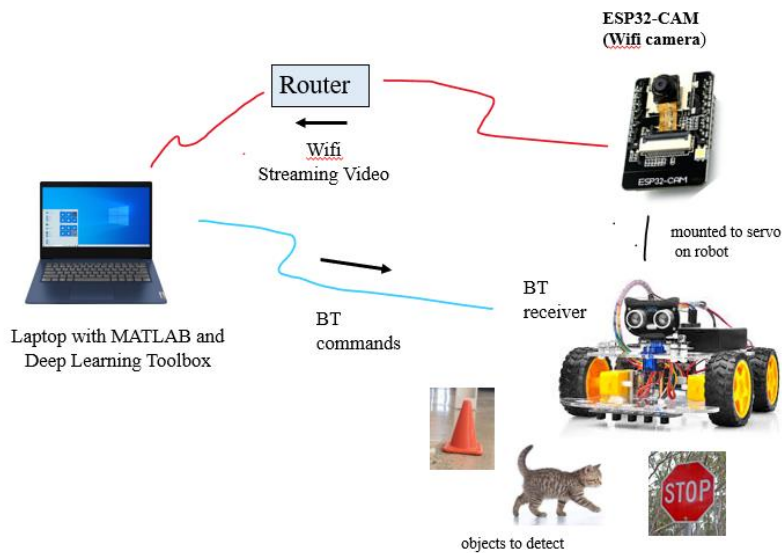
Figure 4: Block Diagram for Deep Learning Integration Project with Mobile Robot

4. Conclusions

Using the proper educational resources, deep learning technology is very accessible tool which can be successfully utilized in projects by undergraduate engineering students with various backgrounds. The paper presented a project-based curriculum with educational strategies used to prepare multidisciplinary engineering design students who possess diverse programming skillsets to be able to design and deploy basic deep learning and computer vision models in relevant application domains in a short time frame and at low cost. The stated learning outcomes were achieved. The preliminary results in terms of overall quality of the module were judged positive as indicated by student feedback. In spring 2021, the course, as specified above, was successfully delivered in Zoom and each student purchased the low-cost Arduino robot and ESP32-CAM wifi camera. During this period of spring 2021 and 2022, 90% of the students were able to successfully complete the required labs during the 7-week module. In both 2021 and 2022, the Yolo object detection lab with robot integration was optional (extra credit) but has been established as a required lab in 2023. Assessment will continue in future installments and delivery of this deep learning module. This paper will hopefully be useful to engineering educators who wish to integrate AI and deep learning exposure into the engineering curriculum, across many disciplines. Due to the importance and widespread use of AI and deep learning in research and industry, it is anticipated that students will benefit from exposure to this exciting and highly effective technology.

References

[1] LeCun, Y., Bengio, Y. & Hinton, G. "Deep learning", *Nature* **521**, 436–444 (2015).

[2] Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* "Review of deep learning: concepts", CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021).

[3] Dinggang Shen, Guorong Wu, Heung-Il Suk, "Deep Learning in Medical Image Analysis", Annual Review of Biomedical Engineering 19:1, 221-248 (2017).

[4] Esteva, A., Chou, K., Yeung, S. *et al.* "Deep learning-enabled medical computer vision", *npj Digit. Med.* **4**, 5 (2021).

[5] Cai L, Gao J, Zhao D. "A review of the application of deep learning in medical image classification and segmentation", Ann Transl Med. (2020).

[6] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, pp. 1-6. (2018).

[7] Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *J Big Data* **8**, 53 (2021).

[8] https://www.mathworks.com/help/deeplearning/gs/get-started-with-transfer-learning.html

[9] ChatGPT, GPT-4 website: https://openai.com/blog/chatgpt; https://openai.com/research/gpt-4

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788 (2016)

[11] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Part of Advances in Neural Information Processing Systems 25 (NIPS 2012).