Paper ID #36068

Integrating Broad Background Content into an Introductory Course on Applied Artificial Intelligence

Dr. Ramakrishnan Sundaram, Gannon University

Dr. Sundaram is a Professor in the Electrical and Computer Engineering Department at Gannon University. His areas of research include computational architectures for signal and image processing as well as novel methods to improve/enhance engineering education pedagogy.

Work-in-Progress: Integrating Broad Background Content into an Introductory Course on Applied Artificial Intelligence

Ramakrishnan Sundaram
Department of Electrical and Cyber Engineering
Gannon University
Erie, 16541
E-mail: sundaram001@gannon.edu

Abstract

This paper discusses the integration of broad background knowledge into an introductory course on applied artificial intelligence. The engineering programs at universities across the world must adapt to the rapidly changing engineering technology and the needs of the global workforce. The engineering students who enroll at these universities expect to be educated and trained with the latest industry-approved tools in order to function effectively in the engineering industry. In recent years, artificial intelligence (AI), machine learning (ML), and deep learning (DL) have burgeoned to successfully reduce or eliminate human interaction while yet processing large amounts of data.

The broad description of the introductory course in applied artificial intelligence is to design physical systems requiring a background in digital signal and image processing, artificial neural network architectures, and specific programming skills in C/C++/Python. The students will have individual and team responsibilities as they function on project teams tackling real-world applications of AI. In addition, possessing the knowledge and skills to design and implement models in the framework of the internet of things is highly valued from the standpoint of assembling a physical system or a product which integrates the functional aspects of traditional subjects with modern tools to deliver tangible outcomes in the real world.

The course also prepares the student to participate on the integrated project platform for co-curricular and research project activities. The integrated platform was designed to test and implement the next generation of intelligent ground vehicles. The platform comprises modules for training data sets using the neural network, performing object detection and classification, followed by collision avoidance. In addition to course and curriculum development, the platform supports the active participation of the student, undergraduate and graduate, in design competitions related to intelligent and autonomous vehicles.

Introduction

The rapidly changing engineering technology and the needs of the global workforce in the 21st century compel engineering programs at universities across the world to adapt their curricula to prepare graduates for the new reality. The adaptation can be the restructure of courses in traditional subjects and/or the adoption of entirely new courses with content tailored to educate and train the student with the latest industry-approved tools thereby preparing each of them to function effectively in the engineering industry. Artificial intelligence (AI), machine learning (ML), deep learning (DL), and the internet-of-things (IoT) have been able to reduce or eliminate human interaction while yet processing large amounts of data [1]-[14]. AI offers computational tools that replace the need for humans to perform certain repetitive tasks. The industries which already use AI include health care, retail, manufacturing, ad banking. The engineering students at Gannon University are taught the set of courses whose content can be integrated into a course on applied AI. The approach in the pilot course is at the level of subsystem and system design, with the underlying theory simply summarized and treated as a black box which receives the appropriate inputs and delivers the desired outputs.

The paper will list the measurable outcomes of the course, the outline of course activities, and the course assessment methods. The course is taught over fourteen weeks with a total of twenty eight class sessions, each of duration eighty minutes. First, the student hones his/her skills in Python with exercises designed to yield results based on the application of the concepts related to image processing and image understanding. The student learns to apply convolution and pooling operations in stages to filter image frames and extract feature maps. Thereafter, the student learns how to design and train the neural network for the targeted application such as object detection and/or recognition and/or classification.

Figure 1 captures the essential subject matter necessary for the course on applied AI to deliver measurable course outcomes. The activities conducted during this course fall in the categories of problem-based, project-based, and self-directed learning. The laboratory and project activities of the course emphasize the integration and testing of physical systems by providing the necessary insight into the building blocks displayed in Figure 1.

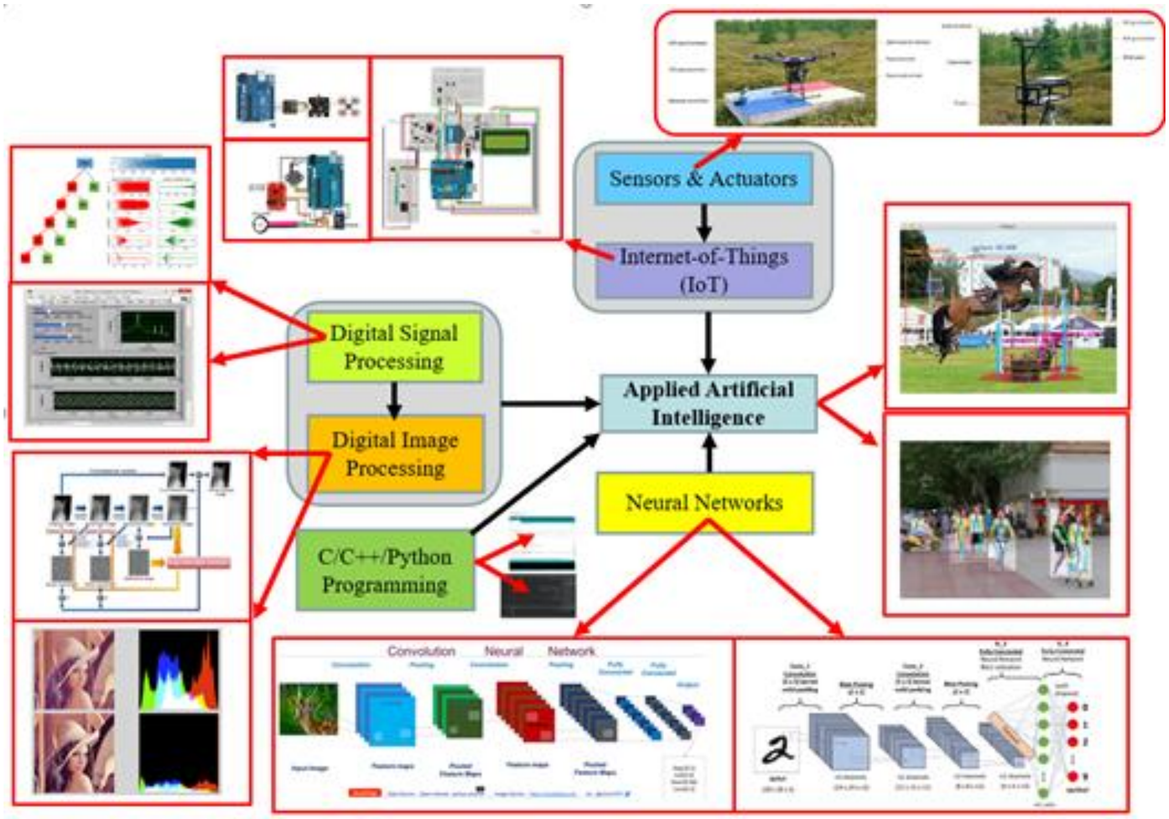


Figure 1: Background preparation

Section 2 overviews the course setup in terms of its description, outcomes, schedule, and assessment methods. Section 3 details the course delivery as a pilot offering in the Spring 2021 semester (fourteen weeks from January to April). Section 4 documents the learning outcomes assessment which comprises mapping to the ABET student outcomes. Conclusions and next steps appear in Section 5.

Section 2: Course Setup

The course will apply artificial intelligence to the design of physical systems. Specifically, the course will utilize image processing techniques, neural network models, machine training algorithms, and deep learning methods to design the systems required to process large data sets. The projects assigned in this course will use software platforms such as OpenCV Python, Keras, Caffe, and TensorFlow. In addition to a background in engineering mathematics from the standpoint of vector-matrix analysis, the students are expected to have experience in the following topics.

Digital Signal Processing:
- Linear and circular convolution in 1-D
- Fourier/wavelet transforms
- Discrete filters in 1-D

Digital Image Processing:
- Convolution in 2-D and 3-D
- Transforms in 2-D and 3-D
- Discrete filters in 2-D and 3-D

Neural Networks:
- Neuron and network models
- Training algorithms
- Applications & implementations

Programming skills:
- *C/C++*
- *Python*
- *Keras, TensorFlow*
- *Caffe*
- *MATLAB*

The course outcomes (COs) are listed as follows:

CO_1: Identify problems where the methods of artificial intelligence apply
CO_2: Select and implement basic and advanced methods of artificial intelligence
CO_3: Design smart systems based on the methods of artificial intelligence

The assessment methods comprise laboratory exercises and projects which integrate the concepts from the topics listed earlier with software implementations using Python, Keras, TensorFlow, and/or Caffe platforms.

The course schedule is shown below in Table 1. There are twenty-eight sessions (two in a week) in the semester. Each session lasts eighty minutes.

Table 1: Course Schedule

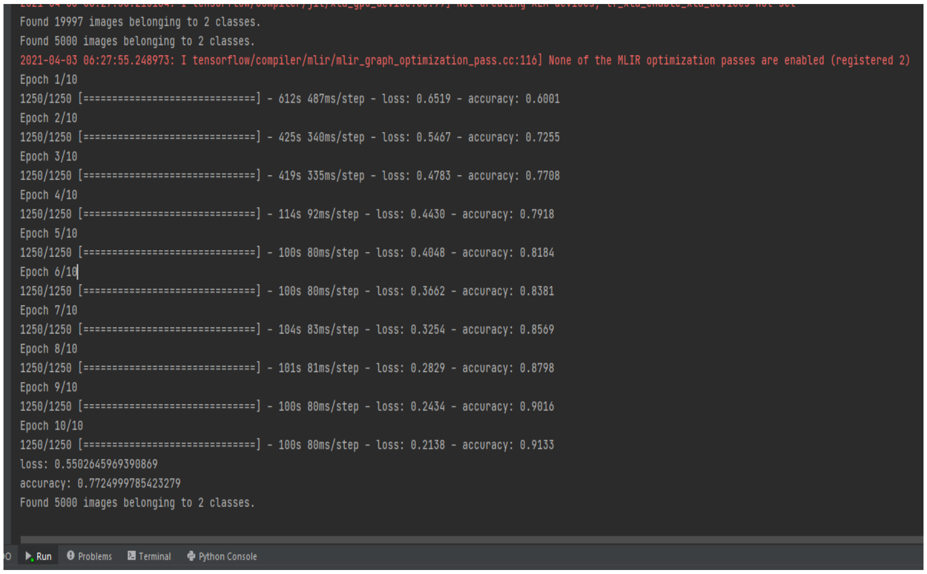
Topic	Week #
Course overview; Image and video fundamentals	1
Laboratory activities - 1: OpenCV Python	2
Laboratory activities - 2: OpenCV Python	3
Project activities – 1: Case studies (Face detection)	4
Project activities – 2: Case studies (Object tracking)	5
Neural Networks (NN) and Machine Learning	6
Mid-term project	7
Convolutional Neural Networks (CNN)	8
Training the CNN for data classification	9
Test the CNN	10
Long-short term memory (LSTM) networks	11
Convolutional LSTM (Conv-LSTM)	12
Applications of LSTM, Conv-LSTM	13
Final project	14

Section 3: Course Delivery

The laboratory exercises during the first six sessions (three weeks) emphasized image processing operations using the OpenCV Python platform. The exercises ranged from image manipulation (translation, rotation, resizing, and cropping), histogram computation to image filtering and edge/contour detection. The first two projects focused on face detection in scenes, video, and real-time webcam feed. The Haar wavelet filter was introduced and implemented as the face detector. The remaining project activities employed neural network architectures for deep learning and are summarized as follows:

Image classification using the CNN

The backbone of feature extraction algorithms is the convolutional neural network (CNN). The application of the CNN architecture to image classification was studied in this project. The student must build and test the CNN to classify a given image. Specifically, the objective is to create the classifier to predict if the given image is that of a cat or a dog. The two stage (convolution and pooling) network was trained and tested in Keras based on a selection of hyperparameters such as number of filters, filter size, batch size, training epochs, and steps per epoch. The convolutional and pooling layers extract spatial characteristics from the images. The fully connected layers, like those in feedforward neural networks, take the spatial characteristics as input and produce the predicted class as the output. Figure 2 illustrates a sample output from the training of the CNN. Figure 3 displays the correctly classified set of 16 randomly tested images.



```
Found 19997 images belonging to 2 classes.
Found 5000 images belonging to 2 classes.
2021-04-03 06:27:55.248973: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)
Epoch 1/10
1250/1250 [==============================] - 612s 487ms/step - loss: 0.6519 - accuracy: 0.6001
Epoch 2/10
1250/1250 [==============================] - 425s 340ms/step - loss: 0.5467 - accuracy: 0.7255
Epoch 3/10
1250/1250 [==============================] - 419s 335ms/step - loss: 0.4783 - accuracy: 0.7708
Epoch 4/10
1250/1250 [==============================] - 114s 92ms/step - loss: 0.4430 - accuracy: 0.7918
Epoch 5/10
1250/1250 [==============================] - 100s 80ms/step - loss: 0.4048 - accuracy: 0.8184
Epoch 6/10
1250/1250 [==============================] - 100s 80ms/step - loss: 0.3662 - accuracy: 0.8381
Epoch 7/10
1250/1250 [==============================] - 104s 83ms/step - loss: 0.3254 - accuracy: 0.8569
Epoch 8/10
1250/1250 [==============================] - 101s 81ms/step - loss: 0.2829 - accuracy: 0.8798
Epoch 9/10
1250/1250 [==============================] - 100s 80ms/step - loss: 0.2434 - accuracy: 0.9016
Epoch 10/10
1250/1250 [==============================] - 100s 80ms/step - loss: 0.2138 - accuracy: 0.9133
loss: 0.5502645969390869
accuracy: 0.7724999785423279
Found 5000 images belonging to 2 classes.
```

Figure 2: Training epochs the CNN for image classification

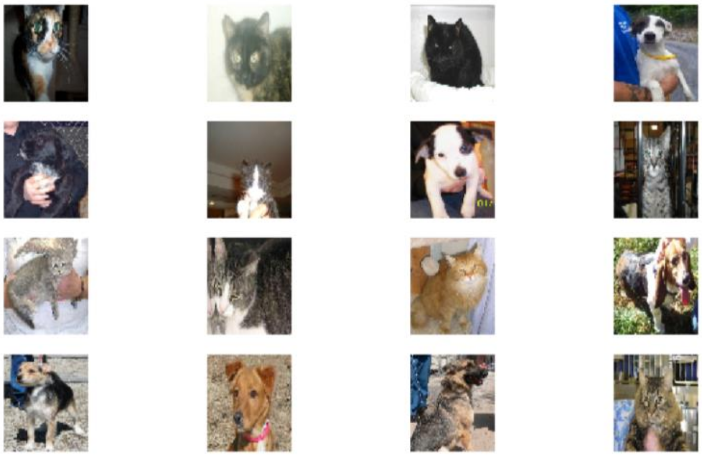
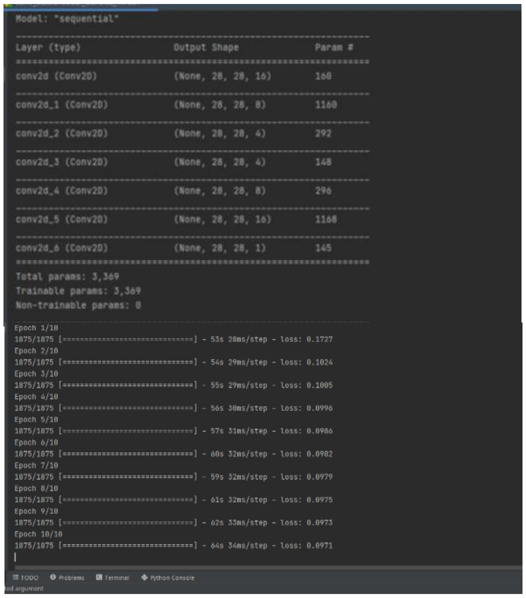


Figure 3: Correct classification

Autoencoders for image compression and denoising

In this project the student must build and test autoencoders to compress and denoise grayscale images. The autoencoders form the class of neural networks which adopt self-supervised learning, i.e., directly from the input as opposed to the CNN which utilizes supervised learning or knowledge of the teacher. The autoencoder has two stages – the encoder to compress the input representation and the decoder to reconstruct the input from the compressed representation. The activities in this project comprise (a) compress grayscale images using one hidden layer with variable node count (b) denoise grayscale images using the CNN architecture in more than one layer of the encoder and the decoder (called deep CNN). The student evaluated the performance of the denoising autoencoder as the depth of the network (encoder and decoder convolutional layers) as well as the filter count in each layer was varied. Figure 4 displays the training phase of the three layer encoder and decoder for denoising images. Figure 5 shows the denoised output images for five randomly selected noisy input test images.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 16)        160
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 8)         1160
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 4)         292
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 4)         148
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 28, 28, 8)         296
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 28, 28, 16)        1168
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 28, 28, 1)         145
=================================================================
Total params: 3,369
Trainable params: 3,369
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
1875/1875 [==============================] - 53s 28ms/step - loss: 0.1727
Epoch 2/10
1875/1875 [==============================] - 54s 29ms/step - loss: 0.1024
Epoch 3/10
1875/1875 [==============================] - 55s 29ms/step - loss: 0.1005
Epoch 4/10
1875/1875 [==============================] - 56s 30ms/step - loss: 0.0996
Epoch 5/10
1875/1875 [==============================] - 57s 31ms/step - loss: 0.0986
Epoch 6/10
1875/1875 [==============================] - 60s 32ms/step - loss: 0.0982
Epoch 7/10
1875/1875 [==============================] - 59s 32ms/step - loss: 0.0979
Epoch 8/10
1875/1875 [==============================] - 61s 32ms/step - loss: 0.0975
Epoch 9/10
1875/1875 [==============================] - 62s 33ms/step - loss: 0.0973
Epoch 10/10
1875/1875 [==============================] - 64s 34ms/step - loss: 0.0971
```

Figure 4: Training of the autoencoder

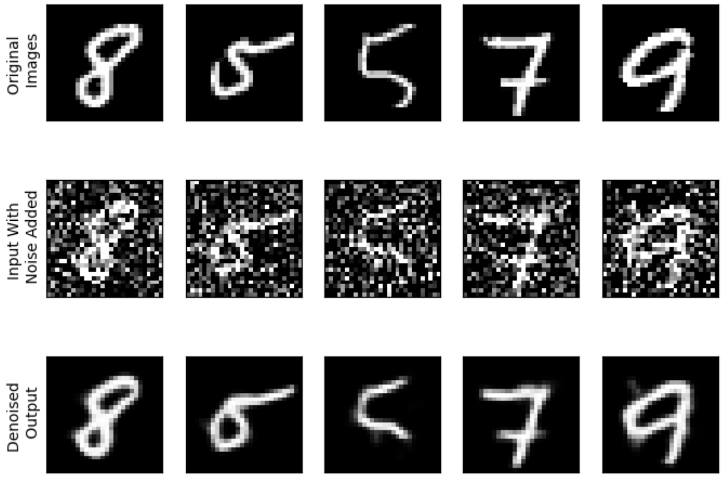
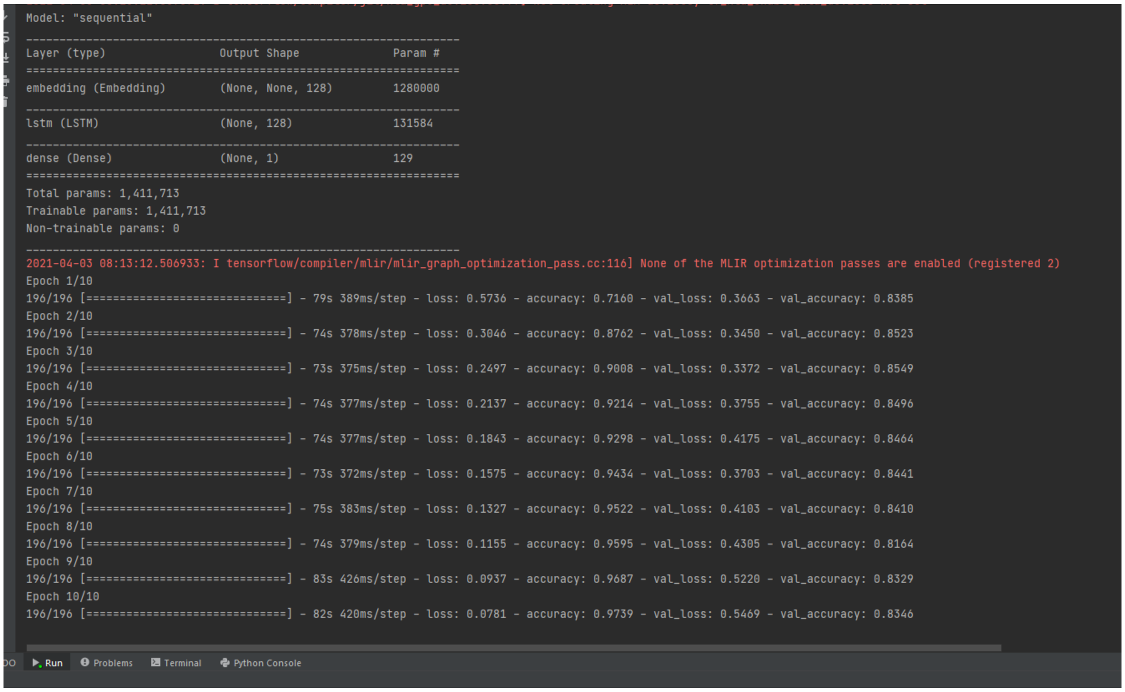


Figure 5: Denoised test output

Recurrent neural networks for NLP

In the class of recurrent neural networks (RNN), the long short-term memory (LSTM) networks apply to sequential problems with time series dependencies. Examples of this are natural language processing (NLP) and video sequences. The project targeted the design and training of an LSTM network to predict the two classes (positive, negative) of movie reviews. The LSTM cell structure is identified and discussed prior to the implementation using Keras. Figure 6 captures the training epochs of the LSTM network.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 128)         1280000
_________________________________________________________________
lstm (LSTM)                  (None, 128)               131584
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 1,411,713
Trainable params: 1,411,713
Non-trainable params: 0
_________________________________________________________________
2021-04-03 08:13:12.506933: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)
Epoch 1/10
196/196 [==============================] - 79s 389ms/step - loss: 0.5736 - accuracy: 0.7160 - val_loss: 0.3663 - val_accuracy: 0.8385
Epoch 2/10
196/196 [==============================] - 74s 378ms/step - loss: 0.3046 - accuracy: 0.8762 - val_loss: 0.3450 - val_accuracy: 0.8523
Epoch 3/10
196/196 [==============================] - 73s 375ms/step - loss: 0.2497 - accuracy: 0.9008 - val_loss: 0.3372 - val_accuracy: 0.8549
Epoch 4/10
196/196 [==============================] - 74s 377ms/step - loss: 0.2137 - accuracy: 0.9214 - val_loss: 0.3755 - val_accuracy: 0.8496
Epoch 5/10
196/196 [==============================] - 74s 377ms/step - loss: 0.1843 - accuracy: 0.9298 - val_loss: 0.4175 - val_accuracy: 0.8464
Epoch 6/10
196/196 [==============================] - 73s 372ms/step - loss: 0.1575 - accuracy: 0.9434 - val_loss: 0.3703 - val_accuracy: 0.8441
Epoch 7/10
196/196 [==============================] - 75s 383ms/step - loss: 0.1327 - accuracy: 0.9522 - val_loss: 0.4103 - val_accuracy: 0.8410
Epoch 8/10
196/196 [==============================] - 74s 379ms/step - loss: 0.1155 - accuracy: 0.9595 - val_loss: 0.4305 - val_accuracy: 0.8164
Epoch 9/10
196/196 [==============================] - 83s 426ms/step - loss: 0.0937 - accuracy: 0.9687 - val_loss: 0.5220 - val_accuracy: 0.8329
Epoch 10/10
196/196 [==============================] - 82s 420ms/step - loss: 0.0781 - accuracy: 0.9739 - val_loss: 0.5469 - val_accuracy: 0.8346
```

Figure 6: Training epochs of the LSTM network

Figure 7 shows the training and validation accuracy of the network with the RMS optimizer.

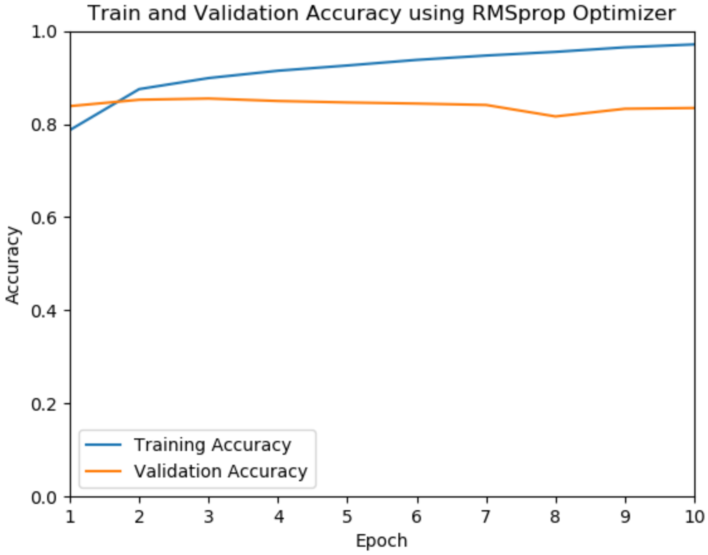


Figure 7: Training and validation accuracy of LSTM network

Object recognition system

The student must build and test the real-time object recognition system to (a) detect objects as in the first project (b) train a neural network to recognize the object in the test image by determining the match with the true image in the database, should it exist.

Section 4: Learning Outcomes Assessment

The learning outcomes assessment comprised the mapping of the course outcomes (CO) to the ABET student outcomes (SO) through performance indicators (PI). Specific course activities called 'Key Assignments' are used to measure the level of achievement of each student based the following scoring: Excellent (E) is scoring 90 or better of the total points possible, Adequate (A) is 75 or better, Minimal (M) is 60 or better, and Unsatisfactory (U) is anything below 60.

- **Identify problems where the methods of artificial intelligence apply (CO_1)**
  PI_4_2: *Understand technology, its application, and potential consequences*

  Key Assignment: Project 1
  Justification: Project 1 requires the student to identify problems which require AI for solutions. The student must understand technology, its application, and potential consequences (PI_4_2). Project 1 measures *the ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts* (SO_4).

- **Select and implement basic and advanced methods of artificial intelligence (CO_2)**
  PI_1_9: *Select and implement the desirable solution and evaluate the results*

  Key Assignment: Project 4
  Justification: Project 4 requires the student to use hardware and software tools to apply AI and deep learning to problem solving. The student must select and implement the desirable solution and evaluate the results (PI_1_9). Project 4 measures *the ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics* (SO_1).

- **Design smart systems based on the methods of artificial intelligence (CO_3)**
  PI_2_5: *Develop systems containing hardware and software components*

  Key Assignment: Term project 1
  Justification: Term project 1 requires the student to design an intelligent system based on AI, neural networks, and deep learning. The student must develop the system containing hardware and software components (PI_2_5). Term project 1 measures *the ability to apply engineering design to produce solutions that meet specific needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors* (SO_2).

The introductory course was first offered in the Spring 2021 semester. The survey completed by each student at the end of the semester comprised the quantitative section and the qualitative section. The quantitative section consisted of specific questions in categories related to the CO (Course Outcomes), assessment techniques, and the overall evaluation of the course.

Course Outcomes

Table 2 displays the responses of the twelve students enrolled in the course. The students are in strong agreement (75%) that the three course outcomes (COs) were satisfied.

Table 2: Course Outcomes

	Outcomes	*Strongly Agree*	*Agree*	*Neutral*	*Disagree*	*Strongly Disagree*	*N.A.*	Mean(5)	**sd**
1	Identify problems where the methods of artificial intelligence apply	75.0	16.7	8.3	0.0	0.0	0.0	**4.7**	**0.31**
2	Select and implement basic and advanced methods of artificial intelligence	75.0	16.7	0.0	0.0	0.0	8.3	**4.8**	**0.21**
3	Design smart systems based on the methods of artificial intelligence	75.0	16.7	0.0	0.0	0.0	8.3	**4.8**	**0.21**
Total Class Response:	**75.0**	**16.7**	**2.8**	**0.0**	**0.0**	**5.6**	**4.8**	**0.24**	

Assessment Techniques

Table 3 displays the survey responses in the category of Assessment Techniques.

Table 3: Assessment Techniques

	Questions	*Highly Ineffective*	*Ineffective*	*Neutral*	*Effective*	*Highly Effective*	*N.A.*	Mean(1)	**sd**
1	Written assignments/Projects	0.0	0.0	0.0	8.3	91.7	0.0	**4.9**	**0.11**
2	Oral presentations	0.0	0.0	0.0	16.7	58.3	25.0	**4.8**	**0.24**
3	Classroom discussion/participation	0.0	0.0	0.0	8.3	91.7	0.0	**4.9**	**0.11**
4	Homework assignments	0.0	0.0	0.0	16.7	83.3	0.0	**4.8**	**0.20**
5	Exams/quizzes	0.0	0.0	0.0	25.0	58.3	16.7	**4.7**	**0.30**
	Total Class Response:	**0.0**	**0.0**	**0.0**	**15.0**	**76.7**	**8.3**	**4.8**	**0.19**

The responses judge the effectiveness of assessment methods used to reflect the knowledge/skills required in this course. The following scale is used to evaluate the effectiveness: 1=Highly Ineffective to 5= Highly Effective. If a particular item does not apply to this course, please indicate "not applicable" = NA. Clearly, at least 75% of the class rated the assessment techniques highly effective.

Overall Evaluation

Table 4 displays the survey responses in the category of Overall Evaluation of the course.

Table 4: Overall Evaluation

	Questions	*Poor*	*Fair*	*Good*	*Very good*	*Excellent*	*N.A.*	Mean(1)	**sd**
1	Overall quality of the course.	0.0	0.0	0.0	25.0	75.0	0.0	**4.8**	**0.27**
2	Overall performance of the faculty.	0.0	0.0	0.0	16.7	83.3	0.0	**4.8**	**0.20**
3	The quality of your learning experience.	0.0	0.0	0.0	16.7	83.3	0.0	**4.8**	**0.20**
	Total Class Response:	**0.0**	**0.0**	**0.0**	**19.4**	**80.6**	**0.0**	**4.8**	**0.22**

The percentage of students rating the course for its overall quality as excellent was 75. At least 80% rated the overall performance of the faculty and their overall learning experience as excellent. The reminder of the class rated the items in the overall evaluation as very good.

Qualitative section

The following comments were captured in the qualitative section of the survey.

1 **Comments:**

- I love the fact that I'm lucky to have my last course in my degree in Artificial Intelligence. Thanks for the department for offering this course and choose the right and creative professor to professionally teach and get the students into real experience in AI.
- Great course.
- it is really good

Section 5: Conclusions and Next Steps

The introductory course in applied artificial intelligence integrates the fundamental and advanced concepts from a broad collection of topics covered in courses such as signal and image processing, neural networks, higher-level programming to enable students to design physical systems for real-world applications such as data classification, object recognition, natural language processing, and video analysis for key point extraction[15]. In the future, the course will incorporate hardware platforms such as the NVIDIA Jetson Nano, TX2, and Xavier modules to perform execute deep learning algorithms on faster and dedicated GPU servers and workstations. The first course did not identify team-based activities as the content of each exercise could be completed by each student individually. However, in future offerings of this course, team-based projects and learning outcomes assessment will be addressed as more complex projects are identified and assigned.

Bibliography

[1] Cristian Iorga et al., "A Deep CNN Approach with Transfer Learning for Image Recognition," *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, June 27-29, 2019*; DOI: 10.1109/ECAI46879.2019.9042173.

[2] Mouna Aftf et al., "Indoor Object C1assification for Autonomous Navigation Assistance Based on Deep CNN Mode," *2019 IEEE International Symposium on Measurements & Networking (M&N), Catania, Italy, July 8-10, 2019*; DOI: 10.1109/IWMN.2019.8805042.

[3] P. Sharma et al., "Deep Convolutional Neural Network Design Approach for 3D Object Detection for Robotic Grasping," *2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV*, Jan. 6-8, 2020; DOI: 10.1109/CCWC47524.2020.9031186.

[4] Z-Q. Zhao et al., " Object Detection with Deep Learning: A Review,"arXiv:1807.05511v2 [cs.CV] 16 Apr 2019.

[5] N. Baker et al., "Deep convolutional networks do not classify based on global object shape," *PLoS Comput Biol 14(12): e1006613, 2018*. https://doi.org/10.1371/journal.pcbi.1006613.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," arXiv:1312.2249v1 [cs.CV] 8 Dec 2013, in *CVPR* 2014.

[7] J. Redman et al., "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.

[8] Juan Du, "Understanding of Object Detection Based on CNN Family and YOLO," IOP Conf. Series: Journal of

Physics: Conf. Series 1004 (2018), DOI :10.1088/1742-6596/1004/1/012029.

[9] F. Particke et al., "Deep Learning for Real-Time Capable Object Detection and Localization on Mobile Platforms," IOP Conf. Series: Materials Science and Engineering 261 (2017) DOI:10.1088/1757-899X/261/1/012005.

[10] D. Yoo et al., "AttentionNet: Aggregating Weak Directions for Accurate Object Detection," arXiv:1506.07704v2 [cs.CV], in ICCV 2015.

[11] R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," arXiv:1311.2524v1 [cs.CV], in CVPR 2014.

[12] Y. Zhang et al., "Real-time vehicle detection and tracking in video based on faster R-CNN," *IOP Conf. Series: Journal of Physics: Conf. Series 887 (2017)* DOI :10.1088/1742-6596/887/1/012068.

[13] K. He et al., "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017; DOI:10.1109/ICCV.2017.322.

[14] Y. Koo et al., "OpenCL-Darknet: An OpenCL Implementation for Object Detection," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Shanghai, China, Jan. 15-17, 2018; DOI: 10.1109/BigComp.2018.00112.

[15] Y. Han et al., "Fusion-based Spatiotemporal Convolutions with Constant Temporal Snapshots for Sign Language Recognition, " *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, December 15-18, 2021.