# Introduction to Autonomous Vehicles (AVs)

* [Safety of Autonomous Vehicles (hindawi.com)](https://www.hindawi.com/journals/jat/2020/8867757/)
* [[1912.09630] Practical Solutions for Machine Learning Safety in Autonomous Vehicles (arxiv.org)](https://arxiv.org/abs/1912.09630)
* [An overview of autonomous vehicles safety | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/8353618)
* [Safety Concept for Autonomous Vehicles | SpringerLink](https://link.springer.com/chapter/10.1007/978-3-662-48847-8_23)
* [Ethics, Safety, and Autonomous Vehicles (miami.edu)](https://repository.law.miami.edu/cgi/viewcontent.cgi?article=2131&context=fac_articles)
* [Design Guidelines on Deep Learning–based Pedestrian Detection Methods for Supporting Autonomous Vehicles | ACM Computing Surveys](https://dl.acm.org/doi/abs/10.1145/3460770)

# Techniques in the field

* [Sensors | Free Full-Text | A Machine Learning Approach to Pedestrian Detection for Autonomous Vehicles Using High-Definition 3D Range Data (mdpi.com)](https://www.mdpi.com/1424-8220/17/1/18)
* [Electronics | Free Full-Text | Deep Learning-Based Pedestrian Detection in Autonomous Vehicles: Substantial Issues and Challenges (mdpi.com)](https://www.mdpi.com/2079-9292/11/21/3551)
* [Pedestrian Detection for Autonomous Vehicle Using Multi-Spectral Cameras | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/8671738)
* [Performance evaluation of CNN-based pedestrian detectors for autonomous vehicles - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S157087052200004X)
* [Fully convolutional neural networks for LIDAR–camera fusion for pedestrian detection in autonomous vehicle | SpringerLink](https://link.springer.com/article/10.1007/s11042-023-14417-x)
* [Design and Evaluation of a Real-time Pedestrian Detection System for Autonomous Vehicles | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/9161768)
* [Pedestrian Detection and Tracking Using Three-dimensional LADAR Data - Luis E. Navarro-Serment, Christoph Mertz, Martial Hebert, 2010 (sagepub.com)](https://journals.sagepub.com/doi/abs/10.1177/0278364910370216)
* [Combining LiDAR space clustering and convolutional neural networks for pedestrian detection | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/8078512)
* [Pedestrian Detection for Autonomous Driving within Cooperative Communication System | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/8886037)
* [Semantic Fusion-based Pedestrian Detection for Supporting Autonomous Vehicles | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/9219723)
* [Pedestrian Detection Using Image Fusion and Stereo Vision in Autonomous Vehicles | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/8661069)
* [A novel visibility semantic feature-aided pedestrian detection scheme for autonomous vehicles - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0140366421002334)
* [06\_Robotica06\_GMonteiro.pdf (psu.edu)](https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=70d23021b1c4e38017a16311652283df525300c3)
* [Vision-based Real-time Pedestrian Detection for Autonomous Vehicle | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/4456404)
* [Deep learning for autonomous vehicle and pedestrian interaction safety - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0925753521003222)
* [Machine Learning Based Pedestrian Detection and Tracking for Autonomous Vehicles | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/10099089)
* [Pedestrian–Autonomous Vehicles Interaction Challenges: A Survey and a Solution to Pedestrian Intent Identification | SpringerLink](https://link.springer.com/chapter/10.1007/978-981-15-0694-9_27)

# MATLAB

* [General MATLAB onramp](https://matlabacademy.mathworks.com/details/matlab-onramp/gettingstarted)
* [Computer Vision MATLAB onramp](https://matlabacademy.mathworks.com/details/computer-vision-onramp/orcv)
  + Pre-req: [Image Processing Onramp | Self-Paced Online Courses - MATLAB & Simulink (mathworks.com)](https://matlabacademy.mathworks.com/details/image-processing-onramp/imageprocessing)
    - Image Segmentation - dividing image into regions of interest
      * Examples: car, lanes, road
        + Techniques:

Edge detection

Segment by texture

Looking for specific shapes v sizes

Segmentation by color (i.e. green screen)

* + - * Binary mask
        + Logical array that indicates region of interest (1 keep, 0 don’t)
        + Length and width is same as original image
        + Can remove bg, identify objects, do calculations, etc.

How:

Intensity thresholding

Create a binary black-and-white image from grayscale by thresholding its intensity values -> mask

* + - * + Text:

Rows of pixels that contain text have more 0 thresholds, whereas rows between lines have more 1s

Rows with text have smaller sums than rows without text

* + - * Improvements
        + Preprocessing image before binarizing

Noise removal

Background isolation and subtraction

* + - * + Postprocessing binary image itself

Emphasize patterns or shapes in binary image

* + - * + Morphological operations can remove or augment features
    - Datastore:
      * A MATLAB var that acts as reference to data source (i.e. folder of image files)
      * MATLAB stores meta information, without importing image data
      * Can import images later when you need them
      * When time to process, read from datastore one at a time, won’t end up with 100+ images in your MATLAB workspace when processing and classifying images
* [Deep Learning MATLAB onramp](https://matlabacademy.mathworks.com/details/deep-learning-onramp/deeplearning)
  + Importing image, then using pre-trained Network to classify said image
  + Datastore!!
    - CNNs in MATLAB work seamlessly with Image Datastores
    - Network can make predictions on collection of images, by providing datastore instead of individual files
    - Augmented data store can pre-process our image data in a line of code, for example, fitting the input size of a pre-trained CNN, making grayscale images have 3 color channels, etc.
  + **Transfer Learning** \*\*\*
    - Why not take a network to modify it to fit our problem and train it on our own images
    - Process of taking pre-trained network, modifying it and re-training it on new data is called Transfer Learning, popular for tackling many deep learning problems
    - Requires much less training than from scratch
    - To perform:
      * Need Network to Train (pre-trained for us to modify)
      * Data to Train with
      * Example images for which we already have the label – supervised learning
      * Specify a set of training options
        + Applying an algorithm that iteratively improves networks ability to identify correct image

Batch size

Max iterations

Learning rate …

From MATLAB Deep Learning course:

To perform transfer learning, you need to create three components:

1. An array of layers representing the network architecture. For transfer learning, this is created by modifying a preexisting network such as GoogLeNet. – or Zoo
2. Images with known labels to be used as training data. This is typically provided as a datastore.
3. A variable containing the options that control the behavior of the training algorithm.

These three components are provided as the inputs to the trainNetwork function which returns the trained network as output.

* + - The most relevant layers are the input and output layers for Transfer Learning
      * To use this network for our data, will need to resize our data
      * Output Layers we’ll most likely have to change
        + For GoogLeNet

Classifier - fullyConnected

Set output size to number of categories in our dataset (new FC layer) to replace old FC layer

Prob - softmaxLayer

* + - * Check that the new architecture is valid prior to training it!!
    - What is a “mini-batch”
      * A subset of the training images (at each iteration), used to update the weights
      * Each iteration uses a different “mini-batch”
      * Once whole training set has been used, that is known as an epoch!
        + Mini-batch size, epoch size, are parameters we can set in algorithm options (3)
    - Goal should always be to minimize the loss function!
* [GitHub list of learning resources for MATLAB](https://github.com/mathworks/awesome-matlab-students?cid=%3Fs_eid%3DPSM_25538%26%01MATLAB+GitHub+for+Students+is+now+available%21%7CLinkedIn%7CPostBeyond)

# Tutorials

* [nuScenes prediction tutorial](https://github.com/nutonomy/nuscenes-devkit/blob/master/python-sdk/tutorials/prediction_tutorial.ipynb)