Final Edge of Poster

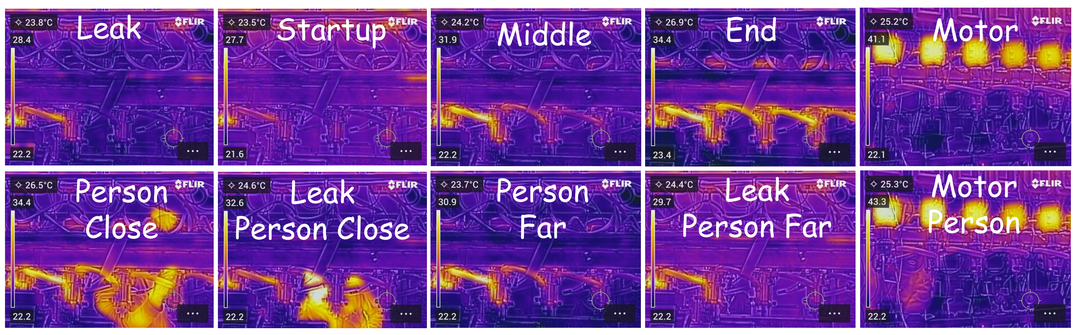
Thermal IR Imagery Analysis Using ML Classifiers & NN for Application to Real-Time Detection

Kelly Truax, Edna Cárdenas, Luis Ocampo Giraldo, Cody Walker, Mitchell Greenhalgh, Katherine Wilsdon, and Jay Hix.

Background

Applications in machine learning (ML) have proven useful for analysis of various types of dataset and real-world problems. Their unique capability towards processing high throughput data with multiple dimensions makes ML especially useful for applications to large volumes of images or video. Data collected from a thermal camera recorded the instance of an unwanted leak in a solvent extraction process. Real-time detection of potential future leaks using ML trained on the known dataset was of great interest and the following goals were proposed: can ML detect the leak from low resolution image/video, can the model identify different classes, and what model would perform best if implemented in real-time.

Classes



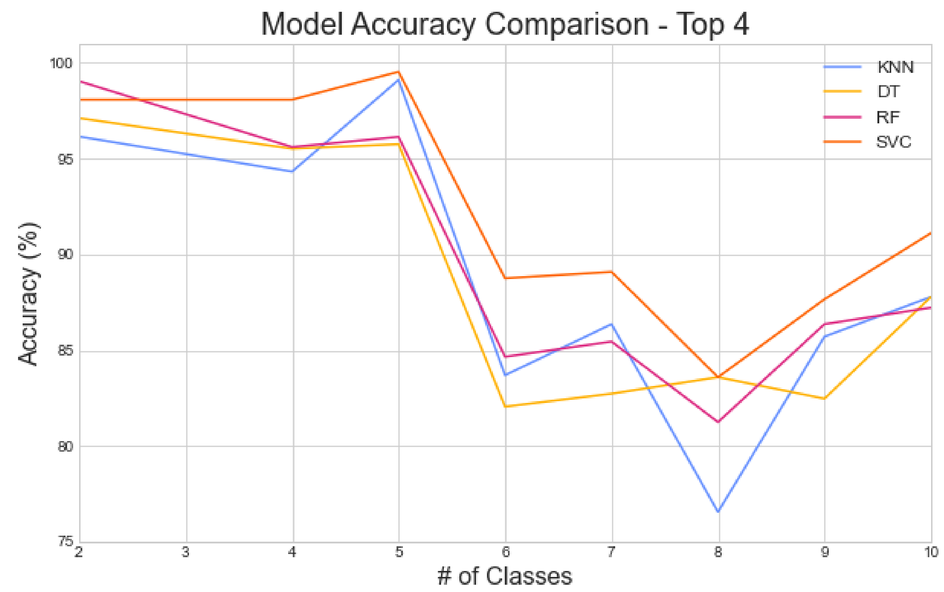
Approach

Of the numerous models available, those tested were chosen based upon applications to image processing in the literature. In total, 10 models were tested, and it was found that linear regression, gaussian, and anomaly detection models were poorly suited to this specific dataset. The 7 remaining models can be divided into supervised classification models and unsupervised neural networks (NN). Each model was constructed, trained, and tested on 12 different combinations of image classes ranging from 2-10 classes for identification. Precision, recall, F1 score, and accuracy were all recorded for each model for each test. Only accuracy is shown for conciseness. No alteration was made to the models between trials save the CNN which changed from “binary” for 2 classes to “categorical” for 2+ classes. Python was used as the coding language and Scikit Learn was used for all models save the CNN which was constructed using TensorFlow and Keras. Models were trained on images extracted from video recordings and classified manually. Train_test_split was used for all models save CNN in which case the training and test sets were manually split 70/30 to match the other models. The best performing model was then applied to image recognition (labeling) and video recordings for user verification.

- KNN – K Nearest Neighbors
- DT – Decision Tree
- RF – Random Forest
- NB – Naïve Bayes
- SVC – Support Vector Classifier
- MLP NN – Multilayer Perceptron Neural Network
- CNN – Convolutional Neural Network

Model Performance

Test #	KNN	DT	RF	NB	SVC	MLP NN	CNN	# of Classes	Image Total
1	100.00	100.00	100.00	91.49	100.00	91.49	100.00	2	155
2	92.31	94.23	98.08	94.23	96.15	94.23	91.84	2	172
3	89.74	91.03	92.31	78.21	96.15	74.36		4	257
4	98.9	100.00	98.90	96.70	100.00	89.01		4	303
5	99.06	91.51	92.28	92.45	99.06	97.17		5	352
6	99.15	100.00	100.00	98.31	100.00	96.61		5	391
7	85.25	81.97	83.61	79.51	91.80	71.31		6	405
8	82.14	82.14	85.71	75.00	85.71	86.90		6	277
9	86.36	84.55	85.45	83.64	89.09	76.36		7	365
10	76.56	83.59	81.25	76.56	83.59	82.81		8	425
11	85.71	82.47	86.36	81.82	87.66	82.47		9	513
12	87.78	87.78	87.22	77.22	91.11	67.22		10	597

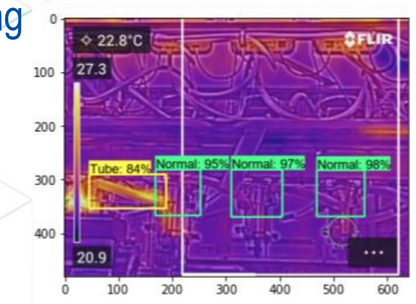


Conclusions

Results of validating multiple models across numerous tests resulted in the SVC having the best performance. The second-best model across all tests was the RF closely followed by the DT and KNN. It appears that the use of NN for this dataset showed no significant benefit over the unsupervised models. Detection of a leak is greatly diminished by the presence of a person in frame. Application of SVC to images and video validated its success in training for identification of multiple classes and its potential for deployment in real-time image classification and problem identification to the operator. Currently an executable file has be implemented to begin classification from external camera input.

Future Work – Object Detection

Further exploration using YOLOv5 and the Object Detection Package for TensorFlow was conducted to discern if the specific location of a leak, or overheating in the case of a motor unit, could be identified. Security concerns supported the stand-alone implementation of TensorFlow, but training could prove challenging for a non-user. More training would be necessary for object detection to outperform the simple classification models, but could be valuable if integrated successfully.



Final Edge of Poster

Not available for content • Not available for content • Not available for content • Not available for content

Not available for content • Not available for content • Not available for content • Not available for content