Meta-Modeling for Efficient Expert Choice in Sample Tube Re-Localization

Johanna Hansen
Jet Propulsion Laboratory, California Institute of Technology
Mobility and Robotics Section
Summer 2019 JPL Internship
Hosted by Renaud Detry

Abstract

NASA is studying concepts for returning physical samples from Mars. One notional aspect of such a campaign would involve a rover that would retrieve tubes containing samples of the Martian terrain that were stored on the surface of Mars in a previous mission. In this work, we consider the problem of developing a vision-based sample tube re-localization capability to facilitate autonomous sample tube grasping and retrieval. Broadly, there are two algorithmic approaches or experts we consider for the task of re-localization. The terrain-relative expert finds a geometric transformation between images collected during sample tube deployment and the rover's current observations, allowing it to find the tube based on human annotations of the deployment image. The direct expert localizes the tube using only current observations. These two experts generally have orthogonal performance cases, with terrain-relative methods failing under large changes in viewpoint, scale, or illumination and direct methods unable to succeed under tube occlusion. Our work seeks to learn to exploit these performance differences by building several experts of each class and a meta-model which learns to identify the most performant expert for a given sample-tube collection assignment. Ideally, our meta-model will be lightweight, robust, and reduce computational cost by allowing us to contextually execute experts. In this report, we demonstrate performance in a variety of state-of-the-art experts and present results with ground-truth localization. In addition, we provide initial meta-modeling results that give insight into the direction of future work on the topic.

1 Introduction

Notional concepts for Mars Sample Return include a Fetch rover that would travel to locations where a previous mission, Mars 2020 for example, deposits hermetically sealed sample tubes, retrieve them and load them onto a Mars Ascent Vehicle (MAV) for eventual return to Earth. Our research group is concerned with developing the capability for the rover to identify individual sample tubes in images and determine their pose relative to the rover position for autonomous grasping and retrieval.

The goal of this summer project was to develop several localization methods for MSR, understand their different failure modes, and eventually develop an efficient meta-model which suggests a localization method (or "expert") for a given sample tube re-localization task.

For the context of this task, we assume that MSR would have the capability to navigate to within 2 meters of the deployment location of each sample tube. Once within the vicinity of a sample tube, the rover would observe the sample tube scene with a Navcam stereo rig. In addition to the observed image containing the sample tube, the MSR rover would have access to an image that the deployment mission captured as each sample tube was deployed. In the up to 10 years between the deployment of

the sample tubes and the MSR mission as currently envisioned, we expect that each of the images captured of the deployment scene would have been annotated by human experts. These annotations are important, as we can leverage human knowledge of the sample tube pose in the deployment image to find the relative tube pose in an observed image of the same scene when we come back to retrieve the tube. Throughout this document, we will refer to *lab-generated*, hypothetical retrieval (MSR) and deployment (M2020) "rover" image observations which were used for researching sample tube localization.

2 Approaches for Re-Localization of Sample Tubes

Broadly, there are two main approaches for re-localizing an object in a scene using machine vision techniques. For the context of this report, we refer to these algorithmic approaches or *experts* as *terrain-relative* (TTL) or *direct-tube* (DTL) localization methods. Each expert is described in detail in the next section.

2.1 Terrain-relative Tube Localization (TTL)

Terrain-relative experts compare the annotated deployment image from tube deployment to the observed retrieval image and determine the relative transformation between the pair. Standard geometric approaches for finding the relative transformation between two images of the same scene follow a multi-stage pipeline.

The first step consists of finding and describing informative features from both images. These features are typically found from algorithmic solutions such as SIFT[7], SURF[1], or ORB[12] or learned from data such as in Deep Desc[13], AffNet[3], or HardNet[8]. Features are typically developed with the goal of being invariant to viewpoint changes such as rotation, scale change, or illumination. Features from both image observations are compared against each other to find try to find feature *matches* that occur in both scenes.

Matches and their location are then used to develop a matrix which describes the transformation between the two observations. With an accurate transformation between the archival and current image, the location of the sample tube can then be found with respect to the MSR vehicle.

(a) Terrain-relative failure case. (b) Terrain-relative success case.

Figure 1: Illustration of failure and success cases for terrain-relative with[9]. TTL methods are fragile when faced with large viewpoint, illumination, or scale changes.

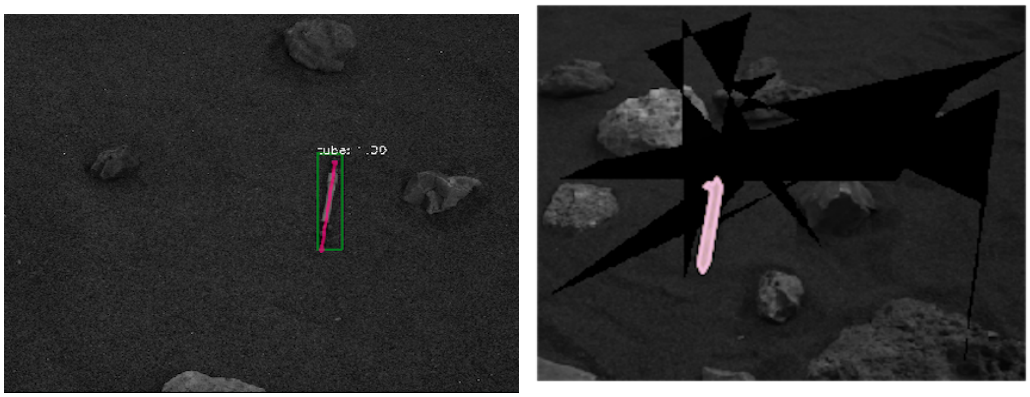
TTL methods have the benefit of being a well-understood approach to solving correspondence tasks, but as we will show later in the document, they can be fragile in scenes where there are few or poor features to match between images, especially when faced with significant scale, illumination, or viewpoint changes (see Figure 1a).

For matching between images, we employ a state-of-the-art open-source implementation of MODS – Matching On Demand with view Synthesis. This efficient matching approach has been shown to work well in wide-baseline matching problems. MODS progressively uses more time-consuming feature detectors, generating synthesized images as needed until a homogrophy is found for a particular image pair. Our early experiments found that MODS performed better than standard OpenCV[2] pipeline and allowed us to easily integrate progressively more performant (slower) feature descriptors. The open-source implementation also enabled us to easily benchmark classic feature descriptors against learned descriptors, respectfully referred to as *Classic TTL* and *Deep TTL* in this report.

In addition to the transformation between the deployment and retrieval images, we also get the number of true and tentative matches from this algorithm. We use the number of true and tentative matches as an approximation to confidence in the transformation estimate.

2.2 Direct Tube Localization (DTL)

Direct localization experts attempt to directly find the tube's pose directly in the MSR image. Object detection and localization approaches have become incredibly powerful in the past few years [15;5;14;6]. Given sufficient data for training, deep neural networks are capable of efficient object detection [11], instance segmentation [5], and pose estimation [6] directly from RGB or RGB-D images.



(a) Mask R-CNN keypoint and object detection

(b) Mask R-CNN mask with shadows

Figure 2

For the direct expert's evaluation, we finetune a state-of-the-art object detection, instance segmentation, and keypoint prediction model, Mask R-CNN [5]. Mask R-CNN extends on its innovative predecessor, Faster R-CNN by adding a branch to the network for predicting an object mask in parallel to the existing branch which predicts a bounding box for each object. Our Mask R-CNN model was trained to predict a bounding box around each sample tube, and then place a mask on the pixels which represent the sample tube with a keypoint for unique endpoints of the tube as shown in Figure 2a.

Mask R-CNN formed a very strong baseline, though we expect it may fail if the observed image is of a sufficiently different distribution than the training set. In an effort to make the DTL method robust to occlusions, we also train it on a dataset with randomly generated shadows, which may resemble those caused by the rover. The model was surprisingly robust to this kind of occlusion as seen in Figure 2b. The bounding box and each keypoint each give us an approximate confidence which can be used.

Our current model does not take into account the rigidity or structure of the sample tube, which sometimes results in a flipped or duplicate endpoint prediction. We think this could be relatively easy to fix in future version of the model.

3 Dataset Development

A balanced set of images that covers both realistic and edge cases for the mission are necessary to properly develop and evaluate our algorithms. The datasets and dataset derivatives used and considered in this project are discussed in Table 3. The DTL method requires example images for training. In order to maintain objectivity of our meta-model, all training images were taken from the "dense" version of the Vicon dataset and all data for the Expert Eval dataset which informs the meta-model were taken from the "sparse" version of the Vicon dataset.

At the time of this writing, we are collecting data in the Mars yard with the goal of adding occlusions (mainly dust/rocks) which make direct localization more difficult. If this still proves too simple for our direct methods, we should investigate simulated worlds where we can add greater difficulty.

4 Meta-Modeling

There are several ways a meta-model architecture could have been developed. In the most naive case, we would always run all of the experts, and then have some meta-model which takes the prediction and confidences given by the experts to choose the true localization estimate. More advanced meta-models may conditionally execute experts and merge their results to find the most precise pose estimation. We ended up designing the model in Figure 3 as our first design, however, we had difficulty getting a satisfactory solution due to an unblanced dataset. In the following few subsections, we describe our process of baselining and learning about expert performance.

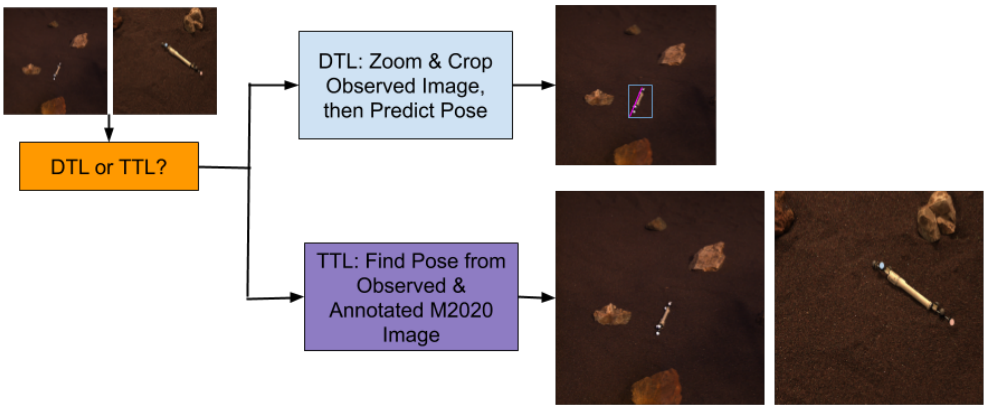


Figure 3: Initial meta-model experts pipeline.

Figure 4

4.1 Predicting an Expert's Performance

We started the meta-modeling problem by asking the question: "Can we predict if a model succeeded or failed, based its prediction and confidence?" One can see how this might be a useful baseline for determining if we are able to trust a model's output before passing this information along to the rover's navigation or grasping mechanisms. We call these experiments which use the expert's own predictions post-facto models, because we actually have to run the expert to get its output and confidence (taking up costly computation cycles).

Our results suggest that we can in fact learn fairly strong discriminators for determining when a model will succeed or fail based on its output using a relatively simple Logistic Regression model, as seen in the third column of Table 1. As expected, we found that balancing classes hurt accuracy, but reduced False Positives (Type I Errors) in Mask R-CNN and had no negative effect on the TTL models as shown in Figure 9. We think this is because the TTL methods generally fail in a very predictable way, and know confidently when they fail, whereas confidence is a bit more ambiguous in the neural model. Adding in normalized keypoint prediction improves performance in all experts, but significantly improves performance on the Mask R-CNN expert dataset. This is because the most common error in the Mask R-CNN expert is that both keypoints get placed on the same end (even though the mask is correct). Results from these logistic regression experiments are found in the *Post-Facto* column of Table 1 and in Figures 6, 7, and 8.

We also addressed a question regarding the relative performance of TTL methods. Given two wide-baseline images of the same scene, can we predict whether a TTL expert will succeed or fail? It is relatively easy for a human with some experience working with geometric feature methods to do this (especially in extreme cases), so we assumed it would probably also be possible for a model to learn it from RGB input. For this experiment, we fine-tuned a pre-trained ResNet18 [4] with the objective of identifying whether Classic TTL would succeed, given a 256x256 retrieval and deployment images as input. Results are shown in Figure 10.

Initial results seen in Figure 10 suggest that the model indeed learns to distinguish image pair homographies which can be successfully solved better than even our post-facto baseline (as shown in the last columns of Table 1). Note that because in this paradigm we don't need to actually run the computationally intensive Classic TTL, we are able to significantly save computation time by not running TTL when it is expected to fail. More investigation into what the model is attending to may provide interesting insight into how to navigate the rover to improve TTL performance.

Expert Class	**Expert**	**% Success**	**% Best**	Post-Facto	Predictive
DTL	Mask R-CNN	**94.4**	**88.1**	**98.7**	
TTL	Deep MODS	31.6	5.0	75.7	
TTL	Classic MODS	21.0	2.9	85.5	88.2

Table 1: Results for each expert on the Expert Eval dataset. Percent Success indicates the percentage of examples in which the expert predicted the tube pose to within a margin of error. Percent Best indicates the percentage of examples in which the expert had the prediction with the lease error. Post-Facto indicates the accuracy of our Logistic Regression model at correctly predicting success or failure of the expert given the expert's prediction and confidence. Input Only indicates the accuracy of a fine-tuned ResNet model which takes in a pair of images and predicts whether or not the expert will succeed.

4.2 Predicting an Expert's Performance

Based on the conclusions from the above section, it seems like we have the tools to effectively predict whether an individual expert will succeed or fail based either on its own prediction and confidence or based solely on the input images. However, what we really want, is to be able to rank a set of experts for a given scene. Unfortunately, the results from this effort have thus far been unsatisfactory.

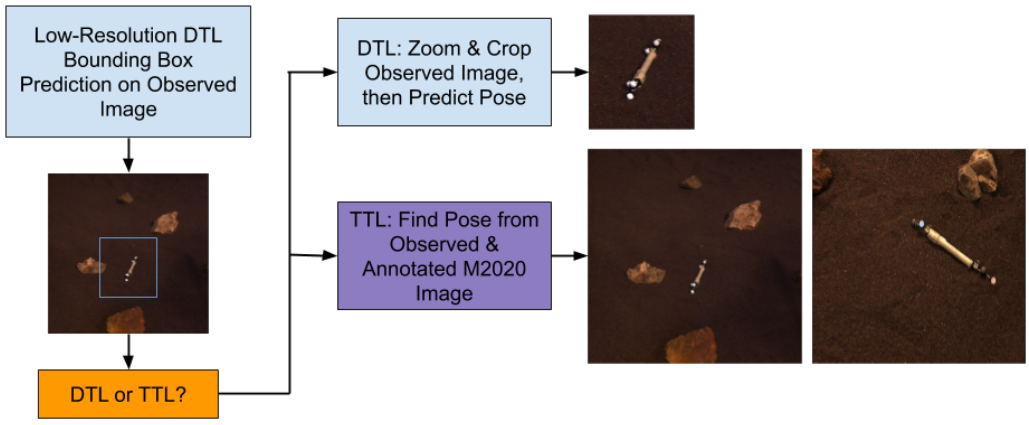


Figure 5: Direct-first meta-model ensemble of experts pipeline

We experimented with running a baseline post-facto SGD classifier[10] to predict the most performant expert, given all of the experts predictions and confidence (the same features used in the success vs. failures task). Due to the high performance of the Mask R-CNN model, our SGD classifier ends up learning a policy of nearly always predicting this expert (confusion matrices included in Figure 11). This results in a high validation accuracy of 94.8%, but low satisfaction, as the model does not learn any of the nuances of DTL failures we'd like to see. Ultimately, we think this is a limitation of our dataset and we look forward to integrating more sources of data.

Ideally, we'd like a lightweight model (such as the one depicted in the pipeline shown in Figure 3) to reduce computational overhead while improving tube localization performance. We trained this model, but though this model outperforms the baseline, it does not capture Mask R-CNN failure cases, as seen in Table 2 and in Figure 12.

Meta-Model	Default	Post-Facto SGD	Resnet Input
Validation Accuracy	93.9	**94.8**	94.5

Table 2: Accuracy of Meta-Models

5 Conclusion

Results from these experiments have led us to rethink the approach we took in Figure 3 in favor of an architecture such as the one seen in Figure 5. In this case, we would always run a lightweight object detector and then feed the output of this model, as well as the simulated MSR and deployment images into a new learned meta model. This meta-model would be tasked with continuing the direct approach, and zooming into the tube for an accurate pose detection, or performing TTL. In this way, we are always running the most performant model (DTL) first on a downsampled version of the full image, before considering switching to TTL, thus saving computation in the likely case that we continue with DTL as the expert of choice.

6 Acknowledgements

The decision to implement Mars Sample Return will not be finalized until NASA's completion of the National Environmental Policy Act (NEPA) process. This document is being made available for information purposes only. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by SPF/SURF program and the National Aeronautics and Space Administration. Many thanks to Renaud, Tu-Hoa, William, and Shreyansh for their knowledgeable suggestions and patient guidance. Thanks to Eric and the intern gang for great conversations and information about navigating JPL.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] J. M. Dmytro Mishkin, Filip Radenovic. Repeatability Is Not Enough: Learning Discriminative Affine Regions via Discriminability. In *Proceedings of ECCV*, Sept. 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

[5] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL `http://arxiv.org/abs/1703.06870`.

[6] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[8] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. Dec. 2017.

[9] D. Mishkin, J. Matas, and M. Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2015.08.005. URL `http://www.sciencedirect.com/science/article/pii/S1077314215001800`.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems 28*, pages 91–99, 2015.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571, Nov 2011. doi: 10.1109/ICCV.2011.6126544.

[13] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative Learning of Deep Convolutional Feature Point Descriptors. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[14] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018.

[15] Z. Zhao, P. Zheng, S. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2019. ISSN 2162-237X. doi: 10.1109/TNNLS.2018.2876865.

A Appendix

Name	Description	Use Case
Vicon	4152 captured images with ground truth tube and camera pose captured in lab with Vicon system with various cameras, lighting conditions, and exposure from spaced viewpoints of two scenes. Rock masks available for 54 camera positions.	This dataset served as the basis for all of the models trained so far in this project.
Expert Eval	140,000 randomly sampled image pairs representing hypothetical MSR (retrieval) and M2020 (deployment) observations of the tube from different viewpoints, distances, and lighting conditions.	Each expert was asked to predict the MSR sample tube pose in each of pairs. A note that we included both "rocks" and "tube" exposure settings in the random selection, but probably should have been consistent in choosing one of the other.
Expert Perf	Output performance of our experts in tube pose position error and uncertainty on our Expert Eval dataset.	This is what are meta-models are trained on.
CFVicon	This dataset augments the Vicon dataset with selectively applied blur, lighting, shadows added to remove features and/or tube from images to induce failures in class of approaches.	A subset of this dataset, with shadows added, was use to test our Mask R-CNN for failures. Although experiments were done demonstrating inpainting and controlled removal of rocks and the tube, the results were not thought to be good enough for training or evaluating the direct method and were ultimately not necessary for inducing failures in the terrain-relative methods.
Rover	Utilize scenes with multiple viewpoints from old rover missions for evaluating our models. Sample Tubes could be randomly dropped in scenes utilizing by finding the ground plane.	Although this may be an interesting dataset to develop in the future, the lack of true ground truth and the difficulty of generating realistic tube poses made the development of this dataset not a priority.
Mars yard	Capture observations from sample tubes in 3 realistic terrains (sparse rocks, gravel, and fractured plates) will be collected in the last week of August. Tubes will be both exposed and partially covered.	This dataset should be useful for evaluating failure of DTL and TTL methods.
Synthetic	Dataset generated several years ago in Blender where a sample tube is dropped in a Mars-like scene and subjected to simulated dust storms.	Could be useful in evaluating and training direct models.

Table 3: Description of datasets used.

Logistic Regression DTL Mask R-CNN Success vs Failure - 98.7 % Valid Accuracy

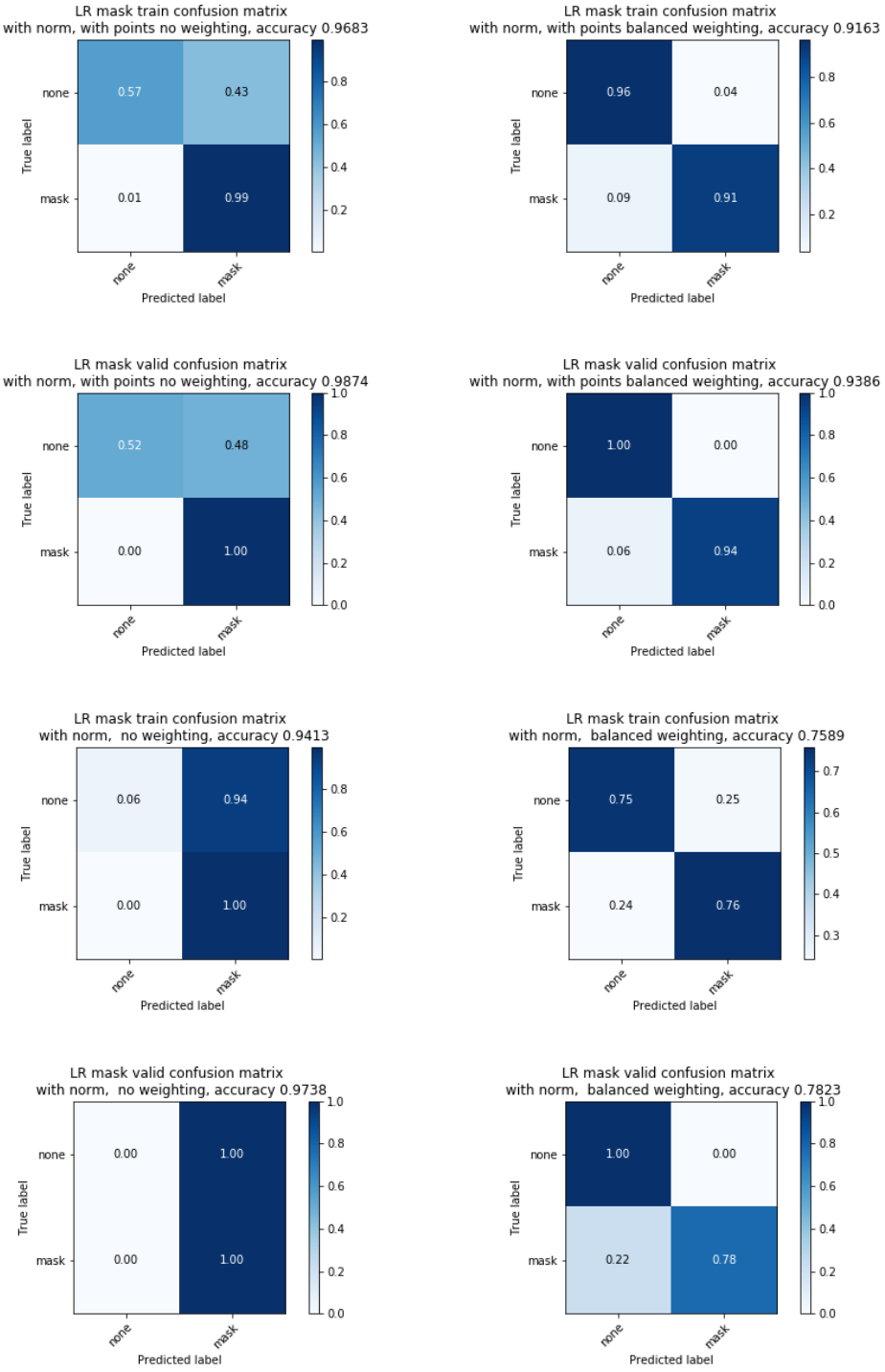


Figure 6: This set of figures depicts the confusion matrices from fitting a baseline Logistic Regression model[10] to predict whether or not a Mask R-CNN expert succeeded or failed to predict the pose of the tube given the confidence output on the keypoints and the normalized represetnation of the keypoint location in the observed image. Results on the training are shown on the first row and results from the validation set are shown on the second row. The right column depicts results when the dataset was re-weighted to induce a class balance, while the left column represents results when the dataset is not weighted. We also include confusion matrices from the training and validation sets on the third and fourth rows when the model is fit without using the keypoint locations.

Logistic Regression TTL Deep MODS Success vs Failure - 75.8 % Valid Accuracy

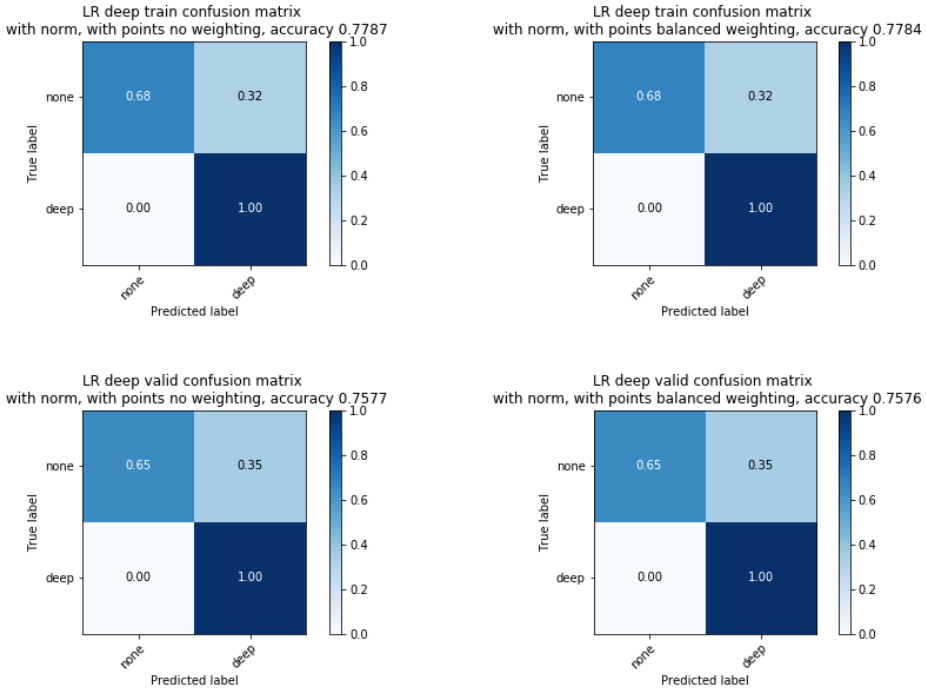


Figure 7: This set of figures depicts the confusion matrices from fitting a baseline Logistic Regression model[10] to predict whether or not a Deep MODS expert succeeded or failed to predict the pose of the tube given the confidence output, the number of true matches, the number of tentative, matches, the normalized representation of the keypoint location in the observed image. Results on the training are shown on the top row and results from the validation set are shown on the bottom row. The right column depicts results when the dataset was re-weighted to induce a class balance, while the left column represents results when the dataset is not weighted. Inclusion of the predicted keypoint location did not significantly improve results.

Logistic Regression TTL Classic MODS Success vs Failure - 85.5 % Valid Accuracy

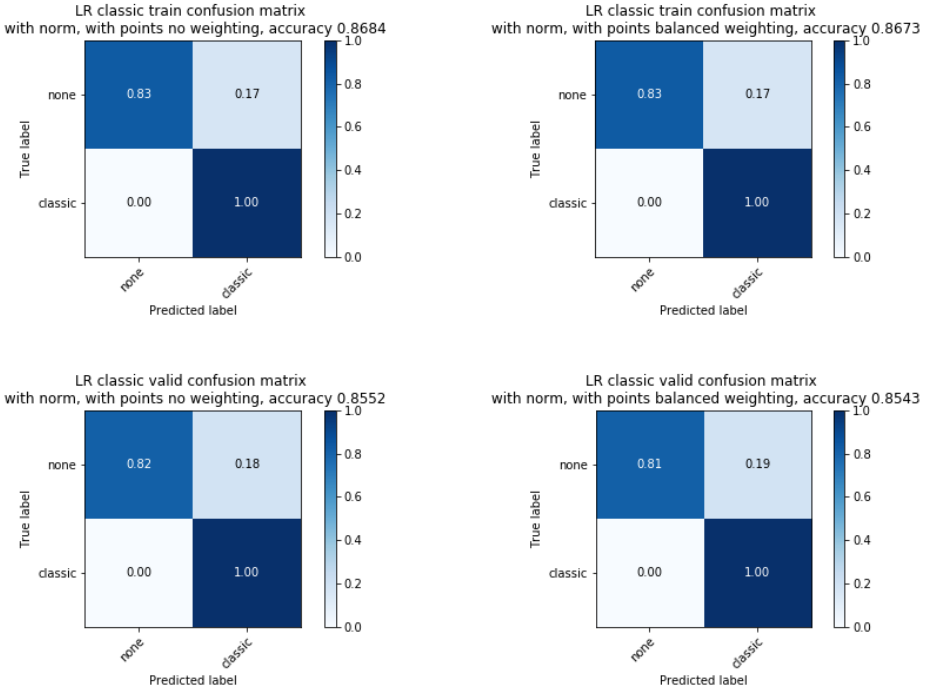


Figure 8: This set of figures depicts the confusion matrices from fitting a baseline Logistic Regression model[10] to predict whether or not a Classic MODS expert succeeded or failed to predict the pose of the tube given the confidence output, the number of true matches, the number of tentative, matches, the normalized representation of the keypoint location in the observed image. Results on the training are shown on the top row and results from the validation set are shown on the bottom row. The right column depicts results when the dataset was re-weighted to induce a class balance, while the left column represents results when the dataset is not weighted. Inclusion of the predicted keypoint location did not significantly improve results.

Which features improve performance in Success vs Failure Logistic Regression

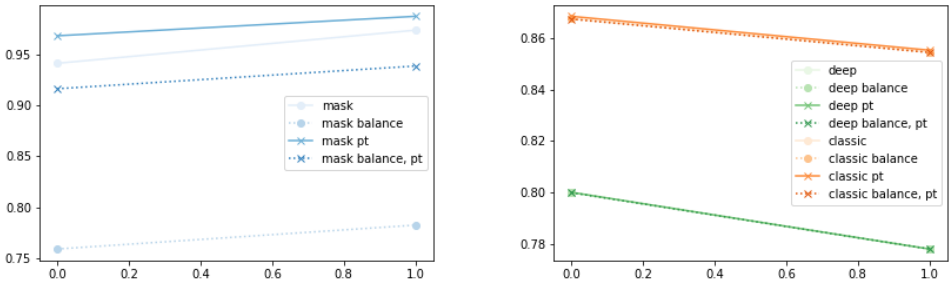
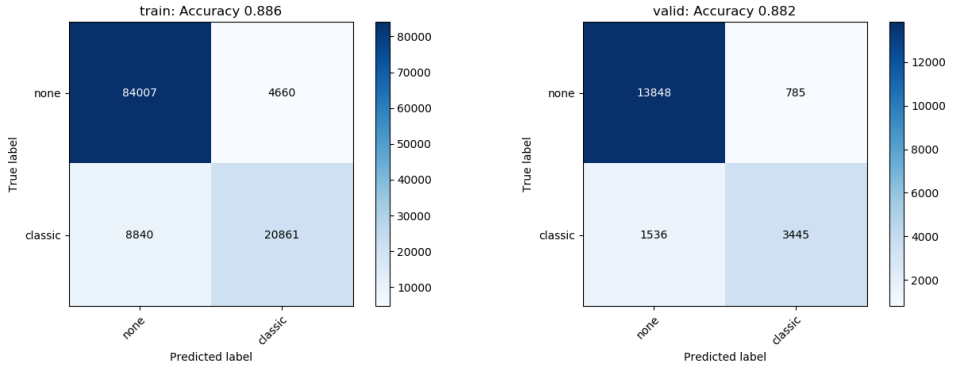


Figure 9: Which features should we include in post-facto success vs. failure models of our experts? Adding the predicted points improved the Mask R-CNN predictions because the most common failure was incorrect keypoints.

Predictive ResNet: TTL Classic MODS Success vs Failure - 88.2 % Valid Accuracy



(a) Training (b) Validation

Figure 10: Results from (partially) fine-tuning a pre-trained ResNet[4] to identify if Classic TTL was likely to succeed using only the image pairs as input (Meta-Success).

SGD for choosing the best expert based on their predictions post-facto 94.8% Accuracy

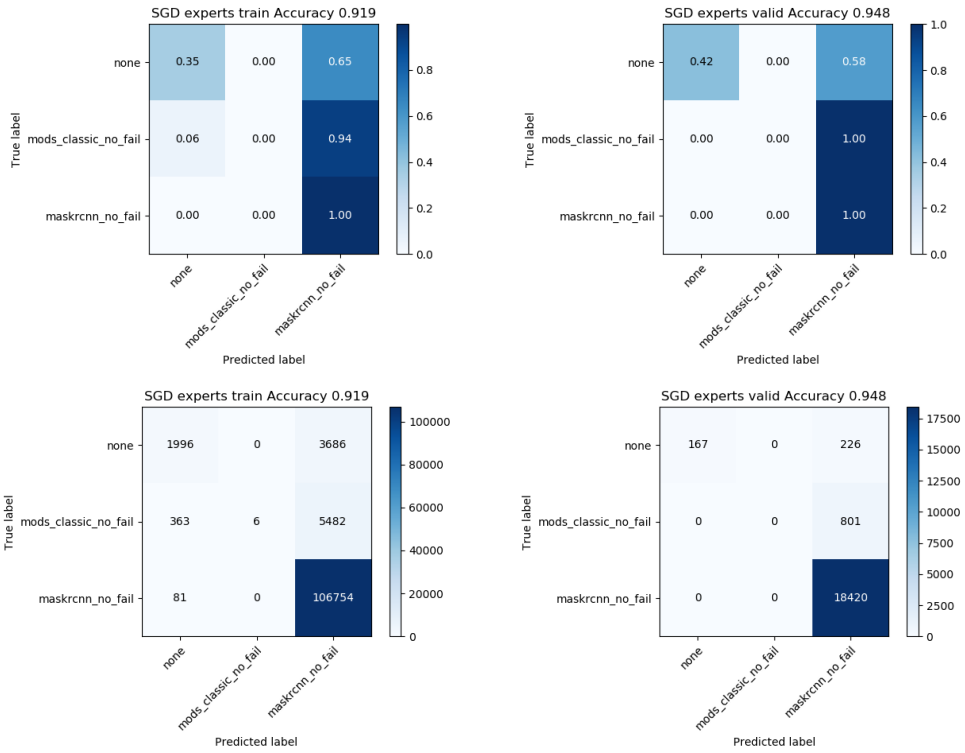


Figure 11: Can we predict the best expert using their own predictions and confidence? The SGD model struggles to overcome the severe class imbalance in our data.

ResNet for choosing the best expert based on input images - 95% Accuracy

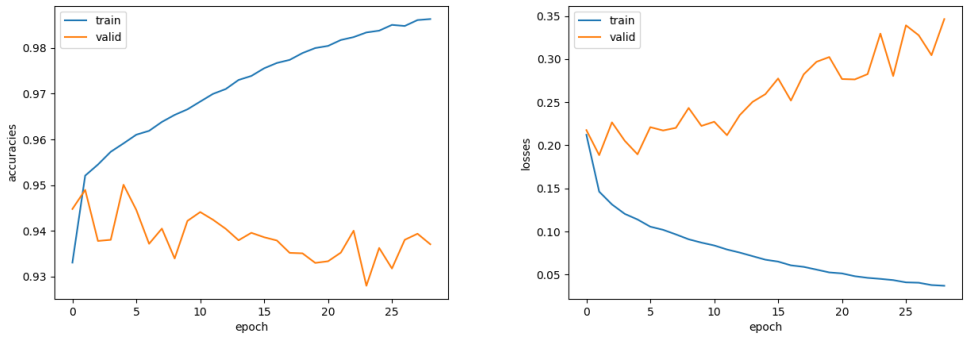


Figure 12: Can we predict the best expert using the staged testbed of the sample tube in a hypothetical MSR image and deployment image? High validation accuracy does not tell the whole story here, as the model learns to only use the DTL method as it dominates our dataset. Early-stopping on the validation set gives us 95% accuracy.