Learning Transferable UAV for Forest Visual Perception

Lyujie Chen, Wufan Wang, Jihong Zhu
Beijing National Research Center for Information Science and Technology (BNRist)
Department of Computer Science and Technology, Tsinghua University, Bejing, China
{chenlj16, wwf14, jhzhu}@mails.tsinghua.edu.cn

Abstract

In this paper, we propose a new pipeline of training a monocular UAV to fly a collision-free trajectory along the dense forest trail. As gathering high-precision images in the real world is expensive and the off-the-shelf dataset has some deficiencies, we collect a new dense forest trail dataset in a variety of simulated environment in Unreal Engine. Then we formulate visual perception of forests as a classification problem. A ResNet-18 model is trained to decide the moving direction frame by frame. To transfer the learned strategy to the real world, we construct a ResNet-18 adaptation model via multi-kernel maximum mean discrepancies to leverage the relevant labelled data and alleviate the discrepancy between simulated and real environment. Simulation and real-world flight with a variety of appearance and environment changes are both tested. The ResNet-18 adaptation and its variant model achieve the best result of 84.08% accuracy in reality.

Videos and Dataset

Additional videos and the full training/testing datasets are available at https://sites.google.com/view/forest-trail-dataset.

1 Introduction

Unmanned Aerial Vehicles (UAVs) have been increasingly popular in many applications, such as search and rescue, inspection, monitoring, mapping and goods delivery. For UAV with very limited payloads, visual technics provide a more feasible way to perceive the world instead of state-of-art radars. In this paper, we primarily study the problem of navigating a monocular UAV in the dense forest by finding a collision-free trajectory, which simultaneously follows the trail and avoids the obstacles.

In recent years, learning based method exceeded the traditional hand-engineered perception and control method. However, because of the wide appearance variability and task complexity, features of clustered forest are much more difficult to extract and learn. As a result, a large amount of data is required to use supervised deep learning. While there has been some progress in navigating drone in forest trail [Giusti *et al.*, 2016], the image dataset provided has so much man-made deviation which leads to wrong image label. Real-time images of the clustered forest with attitude information are expensive and difficult to collect by autonomous navigation. A tiny mistake will lead to the catastrophic and dangerous result. Therefore, acquiring more easily obtained data from simulation environment is an effective alternative. But visual appearance between simulated and real world is not the same. In traditional machine learning method, training and test data are forced to have the same data distribution and input feature space. The learned policies are only put to use in the similar environment and domain that the model was originally trained on. In order to apply policies from simulation to real flight, it is essential to use transfer learning to leverage the relevant labelled data and alleviate the discrepancy between different environments. In general, forest perception and autonomous navigation in the clustered forest is still a challenging task.

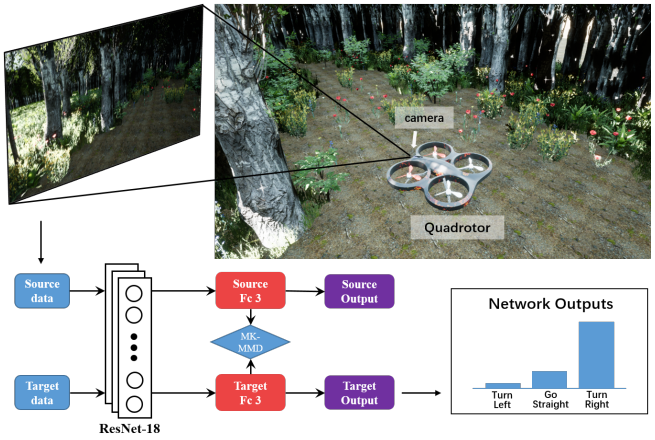
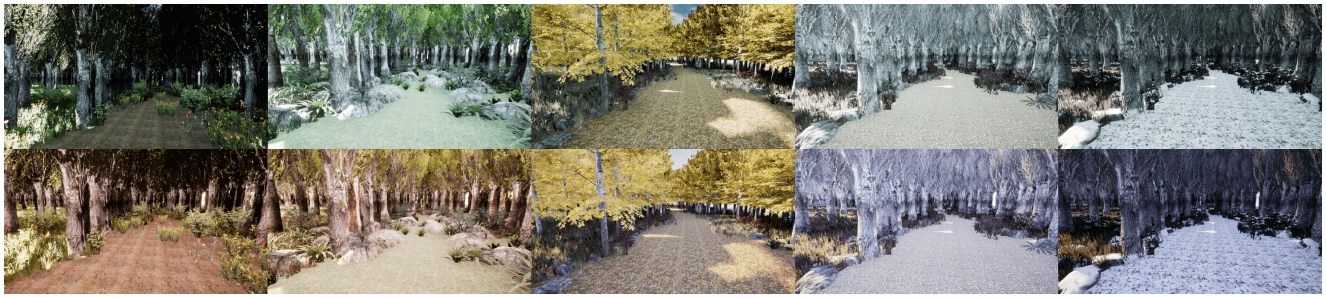


Figure 1: A quadrotor acquires the forest images from a forward-looking camera; a ResNet-18 adaptation network outputs the probabilities of three defined classes, which will be transformed into the control signal of UAV by a simple reactive controller.

In this paper, we solve the problem by first collecting a new dense forest trail dataset in the simulation. Collecting data from the real world is expensive and time-consuming. To retrieve most similar and real pictures of natural scenery,



(a) Scenes in Trail 1

(b) Scenes in Trail 2

Figure 2: Simulated forest trail dataset. From the first column to the last column represent the forest scenes of spring, summer, autumn, winter and snow respectively. For each trail, the first row is the forest in the morning and the second row is at dusk.

we utilize a new simulator named Airsim [Shah *et al.*, 2017] which offers physically and visually realistic simulations built on Unreal Engine to gather a large number of annotated training data in a variety of seasonal conditions and terrain environments. Based on a large amount of data, a deep neural adaptation network is constructed to learn a UAV navigation strategy step by step. Visual perception of forests is viewed as a classification problem. During the flight, proposed model predicts the moving direction to keep the UAV remain on the trail.

The proposed method is validated by a series of experiments, where simulation and real-world flight with a variety of appearance and environment changes are both tested. Among them, we compare the influence of season, lighting, terrain and domain similarity on the effect of transferring improvement. In all tasks, the adaptation model achieves a much better result. It can be observed that different source data affects the adaptation performance to varying degrees. Also, the larger differences between domains, the worse performance achieves on basic model and the greater performance boost on adaptation model. In addition, we propose a new multi-source adaptation model to validate the observations, which gives a very good intuition and a general guideline for making full use of source training data in transfer learning.

The contributions of our work are three folds. (1) A dense forest trail image dataset in a variety of environments. (2) A forest visual perception technique based on deep neural network. (3) A transfer learning technique to control UAV in real world based on the policies learned in simulation environment.

2 Related Work

2.1 Vision Based Forest Perception on UAVs

While Laser rangefinders usually only support 2D detections and radars are too heavy for flight, lightweight vision sensors become a more effective and reliable alternative to perceive the forest. Vision-based techniques on small outdoor UAVs are widely used for obstacle avoidance and path tracking [Giusti *et al.*, 2016; Ross *et al.*, 2013; Daftry *et al.*, 2016b; Dey *et al.*, 2016; Barry and Tedrake, 2015].

Monocular vision techniques are most widely used and attractive because they only rely on a single available camera which is lightweight and easy to deploy. The monocular image can be used as direct input of machine learning [Giusti *et al.*, 2016] and imitation learning method [Ross *et al.*, 2013]. It also can be presented as a depth estimation [Daftry *et al.*, 2016b; Dey *et al.*, 2016].

Stereo systems are always used to compute optical flow and depth map [Byrne *et al.*, 2006; Yang and Pollefeys, 2003]. With more data to be processed, optimization algorithm becomes the crucial part for the real-time flight of stereo vision UAV. Recent work [Barry and Tedrake, 2015] performs an optimized block-matching stereo method to fly a small UAV at over 20 MPH near obstacles and is able to detect the obstacle at 120FPS on an airborne CPU processor.

With the development of artificial intelligent technology, deep learning based methods are widely used in the field of the unmanned system. [Levine *et al.*, 2016] regard the robot control problem as a supervised training process and use a guided policy search to map raw image to robot's motors signals directly. [Giusti *et al.*, 2016] propose a monocular UAV to perceive and follow the forest trail based on a deep neu-

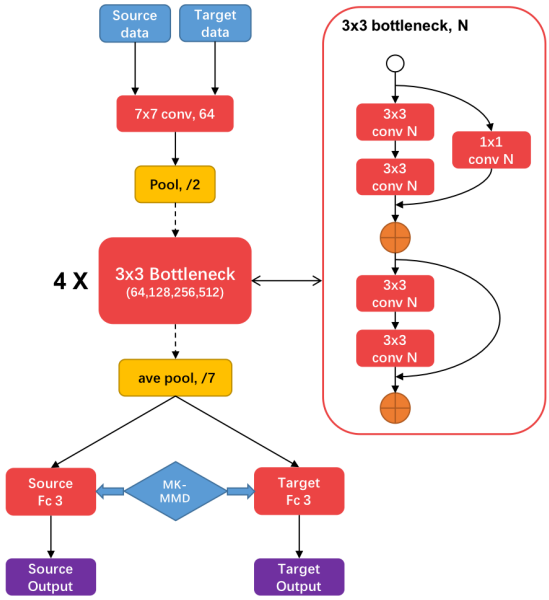


Figure 3: ResNet-18 adaptation network. Convolutional kernel size in all bottlenecks is 3×3 while the filter number goes up from 64 to 512 with every doubling. All convolutional layers are trained via standard SGD. The last fully connected layer is trained to classify the images in the specific domain and should be adapted by MK-MMD.

ral network classifier. It manually collected about 100K images by hiking approximately 7 kilometres forest and mountain trail.

2.2 Transfer Learning

Transfer learning focuses on applying knowledge gained from solved problems to a different but related problem [Weiss *et al.*, 2016]. There are two common ways of transfer learning in the field of deep learning. The most commonly used one is fine-tuning. Many deep learning based methods fine tune from the existing trained model because the target task doesn't have enough data. The other way of transferring knowledge is to change the structure of neural network [Long *et al.*, 2015; 2016]. In this paper, we combine these two methods, which fine tune from a basic learned model while following the deep adaptation network [Long *et al.*, 2015] approach to enhance the ability of autonomous flight in reality.

Deep adaptation network [Long *et al.*, 2015] computes domain discrepancy as the mean embedding of all task-specific layers and adds them to whole network loss. The model jointly improves the performance of source and target. [Daftry *et al.*, 2016a] utilize deep adaptation network for autonomous MAV flight using monocular reactive control. They use Zurich forest trail dataset [Giusti *et al.*, 2016] as source domain and evaluate the effectiveness of proposed method through real-world flight experiments.

Compare with previous works, we gather a more accurate and comprehensive dataset and train a more robust CNN model based on modern transfer learning method.

3 Learning Transferable UAV

In this section, an adaptation network is constructed step by step. We first explain the reason for collecting new dataset and elaborate on gathering a large and representative labelled forest dataset both in the simulated and real-world environment. Then we adopt a ResNet-18 [He *et al.*, 2016] model to perceive the forest by deciding the moving direction for given simulated images. Since the real world has the very different visual appearance, the model learned from simulation hardly performs well in reality. Hence, in order to transfer the learned policies, we further introduce a cross-domain discrepancy metric and propose a new adaptation network. Unlabeled images in target domain are used to enhance the ability of autonomous flight in reality.

3.1 DataSet

In this paper, we do not use the off-the-shelf Zurich forest trail dataset [Giusti *et al.*, 2016] because of the man-made deviation problem. All the images in that dataset were collected by three head-mounted cameras equipped on a hiker. Although the author claimed that they took care of always looking straight along its direction of motion, there were still some destructive and meaningless head-turning and shaking. What's more, the hiker sometimes hesitated when facing sharp turn and not headed straight along the trail. Both of them produced a lot of wrong annotated data.

Collecting data from the real world is difficult and time-consuming. Therefore, there is a need to acquire more easily obtained data from simulation environment. To narrow the gap between simulation and reality, we build variable conditions and environments in Unreal Engine which generate reasonably realistic reconstruction of real-world scenes. We use a simulator called Airsim [Shah *et al.*, 2017] to offer a quadrotor for gathering the images. To carefully follow the dominant direction of the trail and gather high-precision images, we start to make the UAV fly a smooth trajectory in a realistic simulation environment. Next, we walk through the real forest roads by holding the cameras instead of flying a real UAV. We refer to the same acquisition way of [Giusti *et al.*, 2016] by mounting three cameras for easily labelling the data. One pointing straight ahead, and the other two pointing 30 degrees to the left and right respectively. All images acquired by three cameras are labelled. We define three classes, which correspond to three actions that UAV should take to remain on the trail. Specifically, the central camera acquires instances for the class GS(Go Straight). Conversely, all images acquired by the right view camera are of TL(Turn Left) class; and all images acquired by the left-looking camera are of TR(Turn Right) class.

Our forest dataset consists of simulated and realistic parts. The former is composed of 99762 images spread over four seasons, two different trails and several different light conditions as well as viewpoint heights (See Fig.2). The latter is a set of 11103 images composed by 1 hour of 1920 x 1080 30fps video acquired using three portable cameras.

3.2 Deep Neural Network for Forest Perception

Convolutional neural network (CNN) has been shown to perform well in classification problems given a large number of

Task	Training Data Source	Number of Training Data	Validation Data Source	Number of Validation Data	Test Data Source	Number of Test Data
Transfer across season condition	Trail 1 without Winter	41814	Winter of trail 1	300	Winter of trail 1	28005
Transfer across terrain condition	All season of trail 1	70419	All season of trail 2	300	All season of trail 2	29043
Transfer across light condition	All morning data	49752	All evening data	1000	All evening data	49010

Table 1: Transfer Learning Tasks in Simulation Dataset

images [He *et al.*, 2016; Szegedy *et al.*, 2015; 2017]. The problem of finding a collision-free trajectory in the forest can be formulated as the problem of classifying the control action of UAV frame by frame.

In this paper, we adopt a CNN as an image classifier based on ResNet structure. It consists of successive pairs of convolutional and batch normalization bottlenecks, followed by a fully connected layer. In particular, the input layer is considered as a matrix of $3 \times 224 \times 224$ neurons. The input image is resized to 224×224 before mapped to the neurons in the input layer. For a given input, the CNN outputs three values, indicating the probability that the UAV will turn left, turn right and go straight respectively. The training set is only augmented by the horizontal flip to Synthesize the left/right mirrored images. A mirrored training image of class TR (TL) produces a new training sample for class TL (TR). A mirrored training image of class GS still yields a new sample of class GS. Augmentation has double the number of samples.

The model is implemented in Caffe [Jia *et al.*, 2014] and trained using standard backpropagation. Weight parameters $\{W^l\}_{l=1}^{l}$ in convolutional layers are initialized with MSRA filter [He *et al.*, 2015] and all bias parameters $\{b^l\}_{l=1}^{l}$ are initialized to 0. All parameters are jointly optimized using SGD to minimize the misclassification error over the training set (See Eq.1).

$$\min_{\Omega} \frac{1}{n} \sum_{i=1}^{n} J(\theta(x_i), y_i) \tag{1}$$

where $\Omega = \{W^l, b^l\}_{l=1}^{l}$ denotes the set of all CNN parameters among l layers. n is the number of training data. J is the cross-entropy loss function. $\theta(x_i)$ is the output probability of CNN and y_i is the ground truth label given input x_i.

In the test phase, an image from the monocular camera will be fed into the trained model. The output probability corresponds to the control signal of UAV. We implement the same simple reactive controller as [Giusti *et al.*, 2016], which only control the yaw rate and velocity of the flight. The desired velocity is proportional to the probability of GS. The desired yaw rate is proportional to the probability difference between TR and TL, which is P(TR)-P(TL). When the value is positive, UAV is steered to the right. Conversely, a negative value steers the flight to the left.

3.3 Transferable Policy Using Deep Adaptation Network

So far, the learned strategy can only be applied to the simulation environment. However, the real world has very different and variable appearance from the simulation. In this section, we study a problem of transferring our learned policies to real-world flight.

According to the definition of transfer learning, all datasets are divided into two domains. In our problem, the source domain is simulation environment while the target domain is real world, which are characterized by probability distributions p and q, respectively. All data in source domain is labelled, which can be denoted as $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s samples. All data in target domain is unlabelled, denoted as $D_t = \{(x_j^t)\}_{j=1}^{n_t}$ with n_t samples.

In this section, we aim to construct a new deep adaptation network based on the previous ResNet-18 model to bridge the cross-domain discrepancy. The new classifier $y = \theta(x)$ uses the source supervision to minimize the error risk in target domain $\varepsilon_t(\theta) = Pr_{(x,y)\sim q}[\theta(x) \neq y]$.

MK-MMD

Real-world data is hard to collect, which means we have no (or very limited) labelled information in the target domain. To approach this very common challenging problem in domain adaptation, many existing methods aim to simultaneously optimize the performance of source and target domains by introducing a discrepancy metric. In our paper, we use the same measure of domain discrepancy as [Long *et al.*, 2015], which apply the multiple kernel variant of maximum mean discrepancies (MK-MMD) proposed by [Gretton *et al.*, 2012]. The proposed measurement focus on jointly maximize the two-sample test ability while minimizing the Type II error, i.e., incorrectly retaining a false null hypothesis.

Given the domain probability distributions p and q, the MK-MMD $d_k(p, q)$ is defined as the distance between the mean embedding of p and q in reproducing kernel Hilbert space (RKHS). The squared formulation of MK-MMD is denoted by

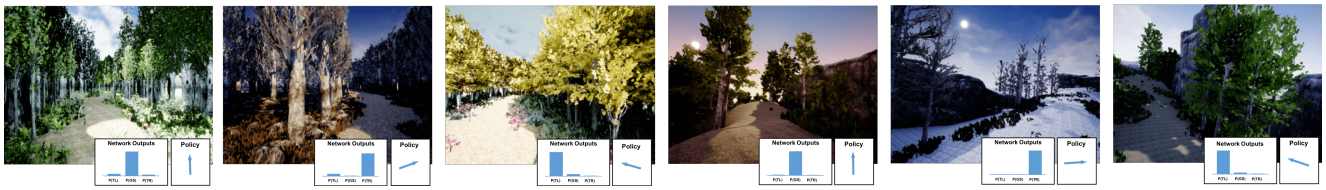
$$d_k^2(p, q) \triangleq \left\| E_p[\phi(x^s)] - E_q[\phi(x^t)] \right\|_{H_k}^2 \tag{2}$$

Where H_k is the reproducing kernel Hilbert space (RKHS) with a characteristic kernel k which correlated with the feature map ϕ. $E_p[\phi(x^s)] =< \phi(x^s), \mu_k(p) >_{H_k}$ where $\mu_k(p)$ is the mean embedding of distribution p in H_k. The most important property of $d_k(p, q)$ is that $p = q \iff d_k^2(p, q) = 0$.

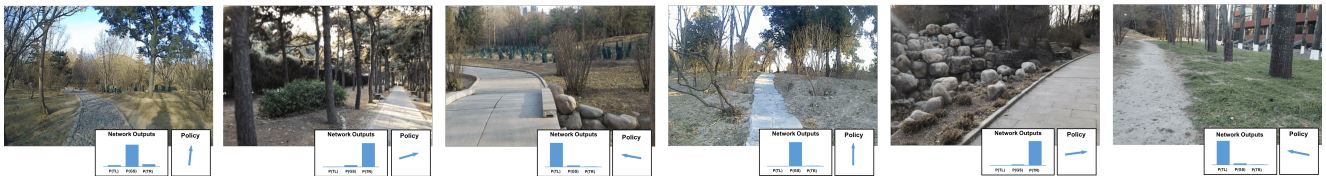
In order to simultaneously minimize the misclassification error and discrepancy between domains, we add an MK-MMD based adaptation regularizer to the fully connected layers of CNN, which introduce the cross-domain discrepancy to basic misclassification error loss (See Eq.3).

$$\min_{\Omega} \frac{1}{n_s} \sum_{i=1}^{n_s} J(\theta(x_i^s), y_i^s) + \lambda \sum_{l=l_1}^{l_2} d_k^2(D_s^l, D_t^l) \tag{3}$$

where λ is a penalty parameter greater than 0, l_1 and l_2 are layer bounds between which the regularizer become effective.



(a) Six frames in simulation



(b) Six frames in real world

Figure 4: Six representative frames in simulation and reality. For each frame, the network outputs and the motion policy are reported based on the ResNet-18 adaptation network.

In our paper, we set l_1=17 and l_2=18, which is the last fully connected layer. Since features always transit from general to specific along the network, the last fully connected layer in our model is trained to classify the images in the specific domain. Hence, the features are not transferable and need to be adapted with MK-MMD.

The new ResNet-18 adaptation network (See Fig.3) is still trained using a mini-batch supervised gradient descent with the above optimization framework (Eq.3). Source and target data are both sent to input layer at the same time. With a large number of unlabeled real-world images, more features of target domain are considered during the training, which will greatly boost the performance.

4 Experiments

4.1 Setup

In this section, we present experiments to analyze the performance of ResNet-18 adaptation network with MK-MMD layer (ResNet-18-Adap) comparing to the basic ResNet-18 network without transfer learning. In order to evaluate the improvement of transfer learning on the tasks, all the experiments are conducted both on ResNet-18 and ResNet-18 adaptation networks. We build three learning tasks on simulation dataset (See table **??**) to evaluate the proposed methods. These are season change test, terrain change test and light change test respectively. Note that, all validation data is randomly retrieved from all data in the target domain. In addition, a learning task of transferring the policy from simulation to reality is built. We compare the effect of different source data on domain adaptation performance.

In all tasks, we first only use data in the source domain to train a ResNet-18 model from vanilla to expert, applying mini-batch SGD with 0.9 momentum and the learning rate annealing strategy. The initial learning rate is set to be 0.05. It requires about 5 hours on a server equipped with an NVIDIA Titan X GPU. Then, we introduce the unlabeled data in target domain to train a ResNet-18 adaptation network. We set its learning rate to be 0.003 and use the SGD with 0.75 momentum. After every 300 iterations of training, a test will be conducted on validation set. We only save the model with the best performance on validation set, then finally evaluate on test dataset.

Transfer Tasks	ResNet-18	ResNet-18-Adap
Transfer across season condition	71.63%	**83.75%**
Transfer across terrain condition	84.25%	**91.33%**
Transfer across light condition	93.23%	**94.33%**
Transfer across simulation and reality	72.24%	**81.40%**

Table 2: Adaptation Results on All Tasks

4.2 Results and Analysis

The supervised learning and unsupervised adaptation results on all tasks are shown in Table **??**. We compare the accuracy of predicting direction with and without adaptation. The performance boost on all tasks indicates that the architecture of MK-MMD adaptation has the ability to transfer learned policies across source and target domains. After that, we apply the trained adaptation model to a UAV in the simulated world. The test video can be found here: https://sites.google.com/view/forest-trail-dataset.

Transfer across simulated environments

In these three experiments, we try to transfer policies over the different simulated environment. The domain shift of season change is induced by the difference in the visual appearance of foliage. While the spring and summer environment is cluttered with dense foliage and the autumn environment has different foliage colour, the characteristics of winter condition are absence of foliage and presence of snow. In this case, the accuracy boost from 71.63% to 83.75%. In the scenario of terrain change, the domain shift is mainly induced by the difference in flight altitude and bumpy trail roads. Comparing to season change, this domain difference is smaller. The basic ResNet model has achieved the great result. The adaptation model reaches 91.33% accuracy rate compared to 84.25% baseline. The domain difference of light change is the smallest. All plants at dusk immerse in the warmer sun-

Source Domain Dataset	ResNet-18	ResNet-18-Adap
All data	72.24%	**81.40%**
All winter data	61%	68.57%
All data without autumn	64%	70.56%
All morning data	**74.59%**	79.74%

Table 3: Performance on Different Source Domain Dataset

Source Domain Dataset	Use Adaptation
All data	81.40%
Four seasons	82.28%
Morning and evening	**84.08%**

Table 4: Multi-Source Adaptation Performance

shine and become yellower. But the other characters are the same. The adaptation model only improves about 1% accuracy rate.

Transfer across simulation and reality

During the reality test, we compare the adaptation performance of four different source domain datasets. At first, we use all simulated data to train a basic ResNet-18 model to mak full use of every feature from all seasons and environments. Then unlabeled real-world data is used to train a ResNet-18 adaptation model. The accuracy boost from 72.24% to 81.40%. Further, we experiment with three more different subsets of all source images in order to figure out which is beneficial for learning (See Table **??**). Since the real world data is gathered in winter, we choose all simulated winter images as the first sub-dataset, trying to extract most of the features in one season. However, the model has very poor performance because the real world forests consist of diverse foliage and terrain compared to clear season appearance difference in the simulated environment. Both evergreen pines and deciduous plants live at the same place. So we choose the second sub-dataset by adding images of spring and summer. At this time, the model has the better result but still worse than the original one. At last, we generate the third sub-dataset as all morning images because the real world dataset has no images at dusk. In this case, the basic model performs better, which demonstrate that the images in evening introduce more cross-domain discrepancy during learning. At the same time, the adaptation model performs worse because more training data provides more features during the transfer learning.

Multi-source adaptation

From the experimental results above, we can make the following observations. (1) In all tasks, adaptation model achieves more than 80% accuracy, which is well enough to make UAV fly automatically. (2) Different source data affects the adaptation performance to varying degrees. (3) The larger difference between domains, the worse performance achieves on basic model and the greater performance boost on adaptation model.

To dive deeper into domain adaptation, we construct a multi-source adaptation model based on ResNet-18 adaptation network to get fully use of different types of features extracted from each source domain (See Fig.5). The simulation dataset is separated based on seasonal and light condition difference (See Table **??**). In these two ways, the accuracies are both higher. Split by light condition, the multi-source adaptation model achieves 84.08% accuracy, which is the best result. Further, we make a sensitive test on learning weight of MK-MMD to investigate the effect of λ. From Fig.6, we can observe that the accuracy first increases and then decreases. When λ equals to 1, the best trade-off between learning forest features and adapting cross-domain discrepancy is achieved.

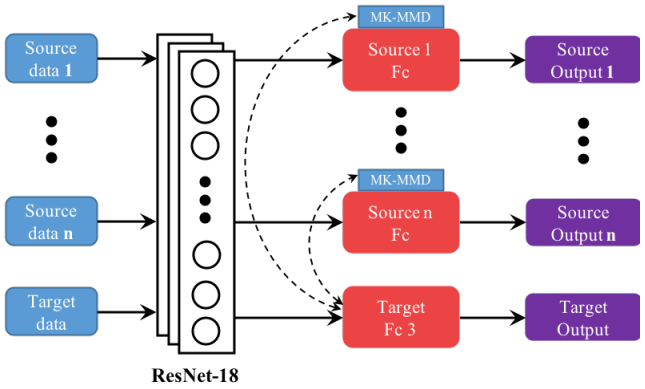


Figure 5: Multi-Source adaptation network. For each source domain, images are fed into the network individually and the corresponding last fully connected layer is adapted by an MK-MMD layer between target domain. To simplify the problem, all source domains use the same learning weight λ.

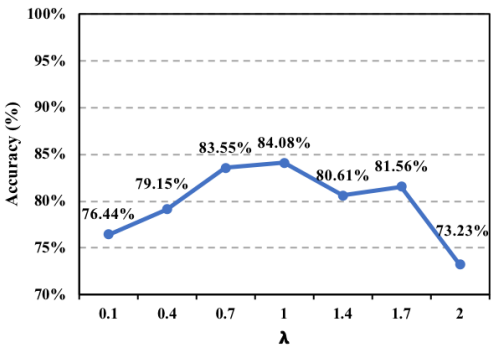


Figure 6: Sensitivity of λ based on multi light conditional adaptation network.

5 Conclusion

We formulate a classification problem to study autonomous navigating a monocular UAV in the dense forest. The navigating strategies are learned in simulation and transferred to the real world to avoid the obstacles and follow the trail. Our new pipeline saves the time of collecting data and reduces the risk of training a real aircraft. In this paper, we propose an adaptation network step by step, which achieves 84.08% accuracy in the real-world test. The MK-MMD layer adapts the task-specific layers to jointly minimize the misclassifica-

tion error and cross-domain discrepancies. Inspired by the result that different types of data transfer specific features to target domain, we propose a simple multi-source adaptation network which achieves the best result. What's more, we implement a simple flight controller and test it in the simulation environment.

One area of future work we plan to address is to construct an end-to-end multi-source adaptation network which can optimize the learning weight of each source domains automatically. In addition, a real flight test is on the schedule. Some engineering improvements should be considered in the future to make the flight stable.

References

[Barry and Tedrake, 2015] Andrew J Barry and Russ Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *Robotics and automation (icra), 2015 ieee international conference on*, pages 3046–3052. IEEE, 2015.

[Byrne *et al.*, 2006] Jeffrey Byrne, Martin Cosgrove, and Raman Mehra. Stereo based obstacle detection for an unmanned air vehicle. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2830–2835. IEEE, 2006.

[Daftry *et al.*, 2016a] Shreyansh Daftry, J Andrew Bagnell, and Martial Hebert. Learning transferable policies for monocular reactive mav control. In *International Symposium on Experimental Robotics*, pages 3–11. Springer, 2016.

[Daftry *et al.*, 2016b] Shreyansh Daftry, Sam Zeng, Arbaaz Khan, Debadeepta Dey, Narek Melik-Barkhudarov, J Andrew Bagnell, and Martial Hebert. Robust monocular flight in cluttered outdoor environments. *arXiv preprint arXiv:1604.04779*, 2016.

[Dey *et al.*, 2016] Debadeepta Dey, Kumar Shaurya Shankar, Sam Zeng, Rupesh Mehta, M Talha Agcayazi, Christopher Eriksen, Shreyansh Daftry, Martial Hebert, and J Andrew Bagnell. Vision and learning for deliberative monocular cluttered flight. In *Field and Service Robotics*, pages 391–409. Springer, 2016.

[Giusti *et al.*, 2016] Alessandro Giusti, Jerome Guzzi, Dan Ciresan, Fang-Lin He, Juan Pablo Rodriguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jurgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 2016.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[Levine *et al.*, 2016] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[Long *et al.*, 2015] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[Long *et al.*, 2016] Mingsheng Long, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. *arXiv preprint arXiv:1605.06636*, 2016.

[Ross *et al.*, 2013] Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert. Learning monocular reactive uav control in cluttered natural environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1765–1772. IEEE, 2013.

[Shah *et al.*, 2017] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[Szegedy *et al.*, 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[Weiss *et al.*, 2016] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

[Yang and Pollefeys, 2003] Ruigang Yang and Marc Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.