WHITE PAPER

Artificial Intelligence
Transfer Learning
Intel AI® Builders

Use Transfer Learning For Efficient Deep Learning Training On Intel® Xeon® Processors

Authors

Beenish Zia
Intel Corporation

Ramesh Illikkal
Intel Corporation

Bob Rogers
Intel Corporation

Table of Contents

Introduction

This is an educational white paper on transfer learning, showcasing how existing deep learning models can be easily and flexibly customized to solve new problems. One of the biggest challenges with deep learning is the large number of labeled data points that are required to train the deep learning models to sufficient accuracy. For example, the ImageNet*[20] database for image recognition consists of over 14 million hand-labeled images. While the number of possible applications of deep learning systems in vision tasks, text processing, speech-to-text translation and many other domains is enormous, very few potential users of deep learning systems have sufficient training data to create models from scratch. A common concern among teams considering the use of deep learning to solve business problems is the need for training data: "Doesn't deep learning need millions of samples and months of training to get good results?" One powerful solution is transfer learning, in which part of an existing deep learning model is re-optimized on a small data set to solve a related, but new, problem. In fact, one of the great attractions of transfer learning is that, unlike most traditional approaches to machine learning, we can take models trained on one (perhaps very large) dataset and modify them quickly and easily to work well on a new problem (where perhaps we have only a very small dataset). Transfer learning methods are not only parsimonious in their training data requirements, but they run efficiently on the same Intel® Xeon® CPU-based systems that are widely used for other analytics workloads, including machine learning and deep learning inference. The abundance of readily available CPU capacity in current datacenters, in conjunction with transfer learning, makes CPU-based systems the preferred choice for deep learning training and inference.

Today, transfer learning appears most notably in data mining, machine learning and applications of machine learning and data mining.[4] Traditional machine learning techniques attempt to learn each task from scratch, while transfer learning transfers knowledge from some previous task to a target task when the latter has fewer high-quality training data.

This paper is divided in to six sections and provides a high-level introduction to the basics of transfer learning, when to use transfer learning, a few real-world applications, a case study showing how transfer learning compares to traditional deep learning, and some areas where transfer learning can be used.

Download Transfer Learning White Paper (PDF).

What Is Transfer Learning

The idea of transfer learning is inspired by the fact that people can intelligently apply knowledge learned previously to solve new problems. For example, learning to play one instrument can facilitate faster learning of another instrument. Transfer learning has gained attention since its discussion in the Neural Information Processing Systems 1995 workshop on *Learning to Learn*[4], which focused on the need for lifelong machine learning methods that retain and reuse previously learned knowledge. Another good analogy is with traditional software development: We almost never write a program completely from scratch; every application makes heavy use of code libraries that take care of common functionality. Maximizing code reuse is a best practice for software development, and transfer learning is essentially the machine learning equivalent.

As described by Bob in his article *Start your data analytics journey today with open source and transfer learning*;[1] transfer learning is an artificial intelligence (AI) practice that uses data, deep learning recipes, and models developed for one task, and reapplies them to a different, but similar, task. In other words, it's a method in machine learning where a model developed for one task is used as a starting point for a model in a second task. Reuse of pretrained models allows improved performance when modeling the second task, hence achieving results faster.

Vast quantities of readily available data are great, but it isn't a prerequisite for success. With modern machine learning and deep learning techniques, knowledge acquired by a machine working on one task can be *transferred* to a new task if the two are somewhat related. This eventually helps to reduce training time significantly, thus improving productivity of data scientists.

There are three main questions to consider when trying to implement transfer learning:

- What to transfer
- How to transfer
- When to transfer

What to transfer asks which part of knowledge can be transferred across domains. Once what to transfer has been addressed the next step is to develop algorithms and models to transfer the knowledge; this falls under the how to transfer step. And the last question asks to which cases should the knowledge be transferred and more importantly not be transferred. In certain situations, brute transfer may even hurt performance of the target task, often referred to as negative transfer.

Transfer learning can be sub categorized in to three types, as mentioned in the paper *A Survey on Transfer Learning*[4] and given in table 1, below.

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	*Inductive Transfer Learning* / *Unsupervised Transfer Learning*	the same	different but related
		different but related	different but related
	Transductive Transfer Learning	different but related	the same

Table 1. Various transfer learning types compared to traditional machine learning, Source[4].

The authors S.Pan and Q.Yang in their paper [4] dive deeper in to the different settings of transfer learning as shown in Table 2, as well as describe different approaches to transfer learning. The four different approaches [4] are:

- instance-transfer, where some labeled data in the source domain is re-weighted for use in the target domain
- feature-representation-transfer, where the model finds a good feature representation that decreases the difference between source and target domains, hence reducing error in regression and classification
- parameter-transfer, where shared parameters between the source and target domain models are discovered and used for transfer learning
- relational-knowledge-transfer, where relational knowledge is built between source and target domains, and data are non-identically distributed as well as represented by multiple relations.

Inductive transfer learning uses all four different approaches, while transductive uses only instance-transfer and feature-representation, and unsupervised transfer learning uses only the feature representation transfer approach. Inductive transfer learning is the most important method for the purposes of this paper. In what follows we will provide examples of inductive transfer learning in both Applications of Transfer Learning as well as in Where Can You Use Transfer Learning, showing

training time speed-up you can achieve using inductive transfer learning.

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Dimensionality Reduction

Table 2. Different settings for transfer learning, Source[4].

Benefits of Using Transfer Learning

What Is Transfer Learning provided an overview of the definition of transfer learning. In this section, we will highlight the benefits of transfer learning while dispelling a couple of deep learning myths.

First myth: You need to have customized hardware infrastructure to do deep learning training. "While in certain cases, training on GPUs tends to outperform training on CPUs, the abundance of readily available CPU capacity makes it a useful platform".[11] This is especially true during the off-peak portions of the continuous inference cycles on the CPU, where idle CPU resources can be utilized for training.[11] Companies like Facebook are currently using dual-socket CPUs for deep learning training for their news feed AI searches,[11] where training frequency is daily and training duration is in hours!

Now that we know CPUs are a great choice for deep learning training, let's get to our second myth: End-to-end training with an enormous dataset is required for deep neural network training. Not always! And transfer learning busts this myth. With use cases like face recognition, where the fundamental features used for classification don't change, there is no need to retrain the complete deep neural network. Transfer learning can be employed in such scenarios, where the features learned using a large dataset are transferred to the new network and only the classifier part is trained with the new, much smaller dataset, as shown in Figure 1. Instead of computing a full forward pass (passing an input image through the entire network to the classification layer) followed by a full backpropagation (using errors to update weights through every layer of the network) on every batch, we just compute the forward pass once using the pretrained model to convert our new training images to feature vectors. We then train on these feature vectors using just a single classification layer. Since we are now training on comparatively low-dimensional data on what is effectively a small model, training completes in minutes rather than hours or days. When doing inference in production, of course we do a full forward pass for each new sample (image).

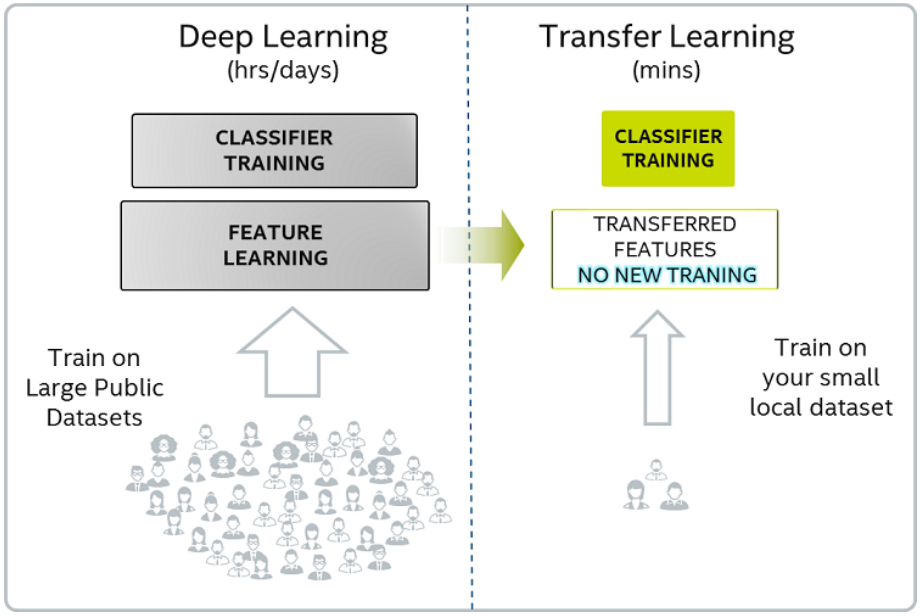

Figure 1. Transfer learning—achieving fast training times with limited dataset.

As the author mentions his article on data analytics and transfer learning,[1] transfer learning, open source deep learning models (often published in a *model zoo*), and standard servers based on Intel Xeon processors can fast track your initial AI programs, and accelerate your return on investment.

Applications of Transfer Learning

Sections What Is Transfer Learning and Benefits of Using Transfer Learning provided a high-level introduction to the concept of transfer learning and its benefits, especially around efficient training using Intel Xeon CPUs. The next thing a reader might want to know is where transfer learning is being used currently. This section gives you a few examples, including research case studies where transfer learning has been applied to solve real-world problems.

More than 465,676 missing children were reported to the Federal Bureau of Investigation in 2016 alone. More than 100,000 escort advertisements are posted online every day, and one in six children reported missing is a possible victim of sex trafficking, as reported by the National Center for Missing and Exploited Children. Intel has worked with Thorn[21] to address the challenge of matching the images of children in the online escort ads with the pictures of known missing children. Thorn is an organization that was able to leverage technology to fight child sex trafficking and apply transfer learning to tackle their huge data challenge.[1,10] Intel helped Thorn take open source models trained on general images of adults and reuse the system to recognize and match images of trafficking victims. To further improve the ability of Thorn to find trafficking victims, Intel used transfer learning on Intel Xeon processors to retrain the model. Using a small dataset of a thousand victims, they took what the algorithm could already do, match general images of adults, and repurpose it to apply it to the new problem.

A similar application of transfer learning is described by Guerra et al. in their paper,[14] where they proposed using transfer learning for real-time sentiment analysis. They used their sentiment analysis approach to analyze the reactions in live microblogs during a game in the Brazilian 2010 soccer season. They transferred the knowledge gained from bias prediction of social media users to their real-time sentiment analysis algorithms. They successfully demonstrated how the knowledge on user bias can be transferred to study overall content polarity in real time, without labeled text data on topics in which polarization occurs. They mention how, "By knowing the bias of only 10 percent of the users who commented about those topics on Twitter, we were able to correctly classify the polarity of 80 percent to 90 percent of the tweets." This approach is different than the recent domain adaption technique of using text-based models to predict document polarity. In the paper, authors state that they've found the transfer learning approach to provide real time sentiment analysis more appealing because of the limited availability of labeled data in comparison to the availability of labeled data in scenarios like product review, and because of the dynamic nature of online discussions, text-based models can become easily outdated. Sentiment analysis in real time, especially for social media, can be useful in many cases. One of the applications that could lead to it being extremely useful is predicting suicidal sentiments in social media posts before taking an adverse action. Detection of such sentiments early on could help save lives.

Another area where transfer learning is gaining popularity is medical image analysis. There have been several research publications in the past few years on how transfer learning is being employed to detect diseases on medical images with a high level of accuracy. In the paper *Transfer representation learning for medial image analysis*[17], two major challenges in automating disease diagnosis are the limited amount of labeled medical data and a lack of domain knowledge experts to identify features in data for detecting target disease. Recently, various researchers have demonstrated the use of transfer learning to address both these problems. In the paper *Understanding the mechanisms of deep transfer learning for medical images*[16], a group of researchers from GE Global Research used transfer learning to detect kidney problems in ultrasound images by transferring a convolutional neural network trained on ImageNet to perform image classification. Similarly, researchers have used transfer learning to detect diseases in medical images [17], where they've used ImageNet, which is the largest image dataset (14 million images over 22,000 categories of everyday objects) to transfer knowledge to their classifier, which is used to analyze otitis media images (any inflammation or infection of the middle ear). What is unique about case studies in both the papers[16, 17] is that instead of using target domain-specific knowledge to extract features from labeled data in that domain, they construct features based on a dataset (ImageNet) entirely irrelevant to the specific domain and still achieve a high level of detection accuracy. This is an example of transductive transfer learning as described in What Is Transfer Learning section of this paper.

In the papers by Zhang et. al.[13] and Burlina et. al.[15] researchers demonstrated the application of transfer learning to detect eye diseases in medical images, with accuracy comparable to human experts. Both papers apply transfer learning to accelerate the diagnosis of age-related macular degeneration (AMD) in medical images of the eye. Figure 2 from the paper on *Identifying medical diagnosis and treatable disease by image-based deep learning*,[13] shows a schematic depiction of how they are using transfer learning to detect specific eye diseases in optical coherence tomography (OCT) images of the retina. In the same paper,[13] the researchers further extended their work to demonstrate the general applicability of such a system for diagnosis of pediatric pneumonia in chest X-ray images. What's remarkable about all these applications of transfer learning in medical image analysis is that this will lead to expediting the diagnosis and referral of treatable medical conditions, resulting in early treatment and improved clinical outcomes.

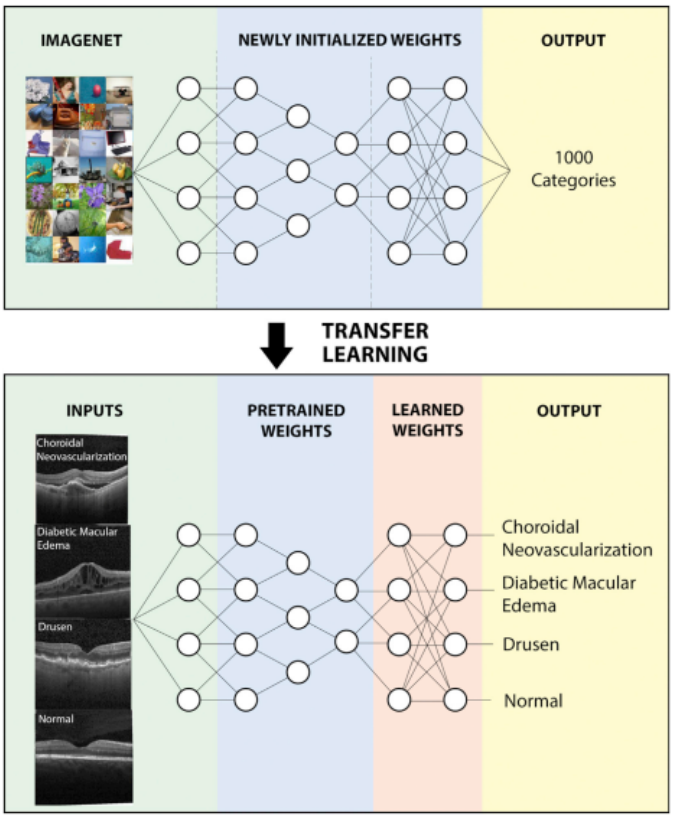

Figure 2. Schematic depiction of application of transfer learning algorithm for analysis of retinal OCT images, Source [13].

There is no dearth of use cases where transfer learning can be applied with a high level of resultant accuracy. Apart from the application of transfer learning in medical image analysis, there have been studies around face verification[19], similar to the use case mentioned in Training Time Data: Traditional Data Learning Versus Transfer Learning on CPU-Based System in this paper, as well as in mispronunciation detection[18]. The fact that transfer learning doesn't need a huge amount of data for training and can repurpose extracted features from a different source task makes it a great technique to apply and use in a variety of domains.

Where Can You Use Transfer Learning

Bob Rogers in his article,[1] states that what we can learn from the Thorn example mentioned in the section above is that it's all about the data, but you don't need to have huge amounts of your own training data. If you take an open source model and retrain it to fit your new challenge, you can quickly have a data analytics solution to solve your problem, without a big upfront capital expenditure, and all made possible through transfer learning.

In Applications of Transfer Learning, we provided a couple of examples where transfer learning is currently being applied successfully. As illustrated above, medical image analysis seems to be a primary area of usage for transfer learning. The lack of labeled data and knowledge experts to label target data makes medical image analysis a great candidate for transfer learning. Another application of transfer learning that has potential to gain popularity is in learning from simulations. In many real-world applications, training models and gathering data is too expensive, time consuming, or dangerous (for example, autonomous driving).[2] Learning from a simulation and applying the acquired knowledge to the real world for such cases would be an instance of transfer learning. Since objects can be easily bounded and analyzed in simulations, learning and gathering data from simulations makes research and development easier. Large-scale machine projects like autonomous driving, which requires vehicles to interact with the real world, can benefit significantly by using simulation data and transfer learning to train and adapt faster.

As mentioned by S. Ruder in his article on *Transfer Learning – Machine Learning's next frontier*,[2] one more application of transfer learning could be applying knowledge learned from one language to another language, especially in the field of automated speech recognition. Transfer learning could be truly beneficial in web document classification. In web document classification,[4] the goal is to classify a given web document into several predefined categories. When trying to classify documents on a newly created website, pretrained classifiers from a different but similar website could be used to transfer the classification knowledge in to the new domain.

In fact, transfer learning has become a best practice for anyone building image classifiers. Even if you have a million labelled samples (and so have enough data to train a really deep model from scratch), you'll still benefit from starting with a pretrained model: a model that has seen both the entire ImageNet dataset and your dataset will outperform a model that has only seen a single dataset. The main difference is that a larger data budget will allow you to fine-tune more layers of the model. With tiny datasets, you risk over-fitting if you retrain more than just the final classification layer, but with larger datasets, you'll see accuracy gains from fine-tuning more layers. A popular approach is to un-freeze one layer at a time, as attempting to retrain a large number of layers can lead to large parameter changes on the first few samples, destroying the very knowledge that you want to transfer.

There is huge potential for growth in the application of transfer learning, given the large innovation around the world happening in AI, the availability of pretrained models, and the uniqueness of users being able to transfer knowledge from one task to another and still getting high levels of accuracy.

Training Time Data: Traditional Data Learning Versus Transfer Learning on CPU-Based System

In the previous sections, we provided an overview on the basics of transfer learning and some good use cases, as well as potential application areas. Now, let's look at the

training time improvement you achieve using transfer learning on an Intel Xeon CPU-based system. This section provides a comparison between training times for traditional deep learning versus transfer learning on the same dataset running on a non-optimized Intel Xeon processor-based system. We didn't have enough data (1000s only) to train a full system, so it's an example of training on a small dataset, as described in the beginning of this paper.

The data in Table 4 below was generated using FaceNet, an image recognition system that directly learns mapping from face images to a compact Euclidean space, where distances directly correspond to a measure of face similarity[7]. The steps described by David Sandberg in his GitHub* publication[8] were followed to run FaceNet and produce the data shown in Table 4. The hardware and software used for collecting data is shown in Table 3. There were no CPU optimizations done for the run. Labeled Faces in the Wild (LFW)[6] was used to train and test the model, when doing both traditional deep learning and transfer learning.

Hardware and Software Bill of Materials

Item	Manufacturer	Model or Version
Hardware		
Intel® Server Chassis	Intel	R1208WT
Intel® Server Board	Intel	S2600WT
(2x) Intel® Xeon® Scalable processor	Intel	Intel® Xeon® Gold 6148 processor
(1x) Intel® SSD 1.2TB	Intel	S3520
Software		
CentOS* Linux* Installation DVD		7.3.1611
Intel® Parallel Studio XE 2018 Cluster Edition		2017.4
TensorFlow*		setuptools-36.7.2-py2.py3-none-any.whl

Table 3. Hardware and software bill of materials used for data collection.

Some of the parameters used for the run

`--max_nrof_epochs=80` (number of epochs run)

`--batch_size = 1024` (number of images to process in a batch)

`--epoch_size = 13` (number of batches per epoch)

LFW dataset[9] has 13,232 images; there are 1024 images per batch, so 12.92 batches, which gives us 13 batches per epoch.

Use Case	Training Dataset	Traditional DL Training Time	Transfer Learning DL Training Time	Speed-up Because of TL	Architecture
FaceNet[7]	LFW (13232 images)	approximately 20 hours	approximately 15mins	approximately 80x	Inception ResNet v1

Table 4. Training time data measured as on a 2S Intel® Xeon® Gold processor-based system, with no CPU optimizations.

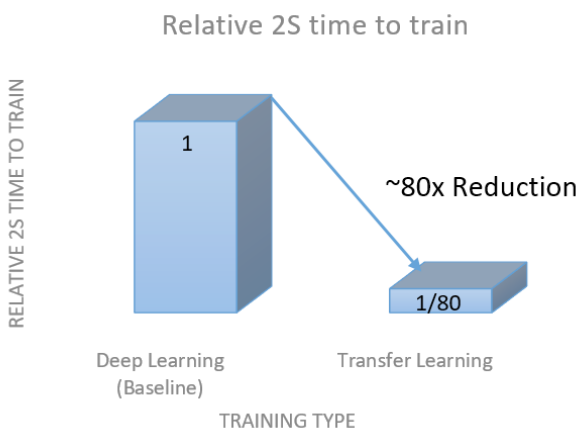

Figure 3. Bar chart showing how transfer learning reduces the deep learning training time drastically, allowing your Intel inference hardware to be reused for training efficiently.

Conclusions

AI has the potential to revolutionize the world. There has been significant growth in the number of application domains where AI can be applied. There are a variety of ways to get started on AI, and this paper describes one of them, transfer learning. Transfer learning eliminates the need for specialized hardware and large datasets, making it easier and faster for users to deploy AI workloads. By using transfer learning, developers can use their current Intel Xeon processor infrastructure with a limited amount of data and start their AI journey today. Transfer learning drastically reduces the training time as seen in the section above. Applying transfer learning to the current Intel Xeon processor inference infrastructure will result in efficient use for the same hardware for training and inference. The same CPU-based infrastructure for deep learning training and inference will provide a more power and cost-effective solution. We expect transfer learning to be applicable to various domains where the learned features do not change (thanks to the rules of nature) and can be reused across domains and problems. In the future, transfer learning techniques will potentially be applied to video classification, social network analysis, and logical inference.

We encourage the readers to further explore the applicability and benefits of transfer learning. For those interested in trying it, we would recommend the FaceNet implementation provided by David Sandberg in his GitHub publication:[8,22] this was the code used by the authors to generate the results shown in Training Time Data: Traditional Data Learning Versus Transfer Learning on CPU-Based System

Acknowledgments:

The authors would like to extend their gratitude to of the many Intel colleagues who reviewed and provided their feedback on this paper. This work would not have been complete without the support of various technology experts at Intel.

References

1. https://itpeernetwork.intel.com/transfer-learning-data-analytics-ai/
2. http://ruder.io/transfer-learning/
3. https://machinelearningmastery.com/transfer-learning-for-deep-learning/
4. https://pdfs.semanticscholar.org/a25f/bcbbae1e8f79c4360d26aa11a3abf1a11972.pdf
5. ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf
6. https://github.com/davidsandberg/facenet/wiki/Validate-on-LFW
7. https://arxiv.org/abs/1503.03832
8. https://github.com/davidsandberg/facenet
9. http://vis-www.cs.umass.edu/lfw/
10. https://software.intel.com/en-us/articles/finding-missing-kids-through-code
11. https://research.fb.com/wp-content/uploads/2017/12/hpca-2018-facebook.pdf
12. https://software.intel.com/en-us/articles/use-tensorflow-for-deep-learning-training-testing-on-a-single-node-intel-xeon-scalable
13. http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5
14. https://dl.acm.org/citation.cfm?id=2020438
15. https://www.sciencedirect.com/science/article/pii/S0010482517300240
16. https://arxiv.org/abs/1704.06040
17. http://infolab.stanford.edu/~echang/HTC_OM_Final.pdf
18. https://www.sciencedirect.com/science/article/pii/S0167639314001010
19. http://openaccess.thecvf.com/content_iccv_2013/papers/Cao_A_Practical_Transfer_2013_ICCV_paper.pdf
20. http://www.image-net.org/
21. https://www.wearethorn.org/
22. https://github.com/davidsandberg/facenet/wiki/Train-a-classifier-on-own-images