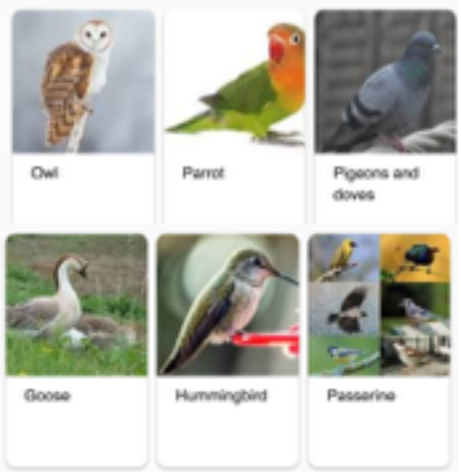
Training Data Synthesis

Our synthetic training data are created using a variety of proprietary methods, Labeling and semantic segmentation are automatic and 100% accurate. We are confident the future of training deep learning system lies is training data synthesis. Here is why

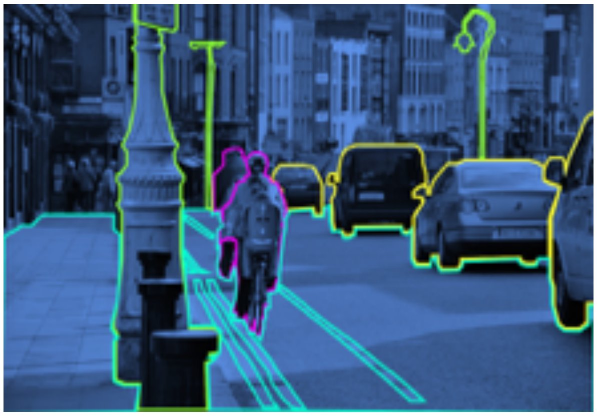
Data collection for building an AI application is a huge challenge. Take, the task of building a system for identifying the species of birds in parks so that endangered species can be protected and visitors can learn more about the birds that are visiting in that season. There are online publicly available databases of various bird species (Fig 1-1).



However, there are very few images of birds sitting on Trees (Fig 1-2).

You actually need to have lots of images of birds on trees to make a viable application. So how do we get them?

Labelling: Even in cases where data collection is not a problem, it gets impossible to label all the data we collect. Take, self driving application - easily 1TB+ of unlabeled data can be generated per day by putting a camera on vehicles that are driving around. but it is virtually impossible to label (Fig 1-3) even if we put an army of humans to do so.



Productivity: AI systems are very expensive to train and need high end GPU's and even clusters. In the beginning however the AI scientist spends time trying to figure out which model, parameters, etc and to iterate they can't work with huge data. TDS dramatically improves the productivity on an AI scientist. Now by just issuing a command, you can get the type of training data you want to verify. For instance, an AI scientist wishes to build a warehouse audit application. He has few ideas on techniques (maybe a fast R-CNN will work, or perhaps transfer learning on a ResNet, or maybe a hybrid of the two?). Using TDS he can define a scene, and then ask TDS to generate lots of variations and try his first hypothesis. He can review the failures and then ask TDS to introduce variations like “put multiple parts in this scene” and then see how his model performs. This can allow him to get the right model, it’s parameters, etc right while another team is working on labeling the actual data.

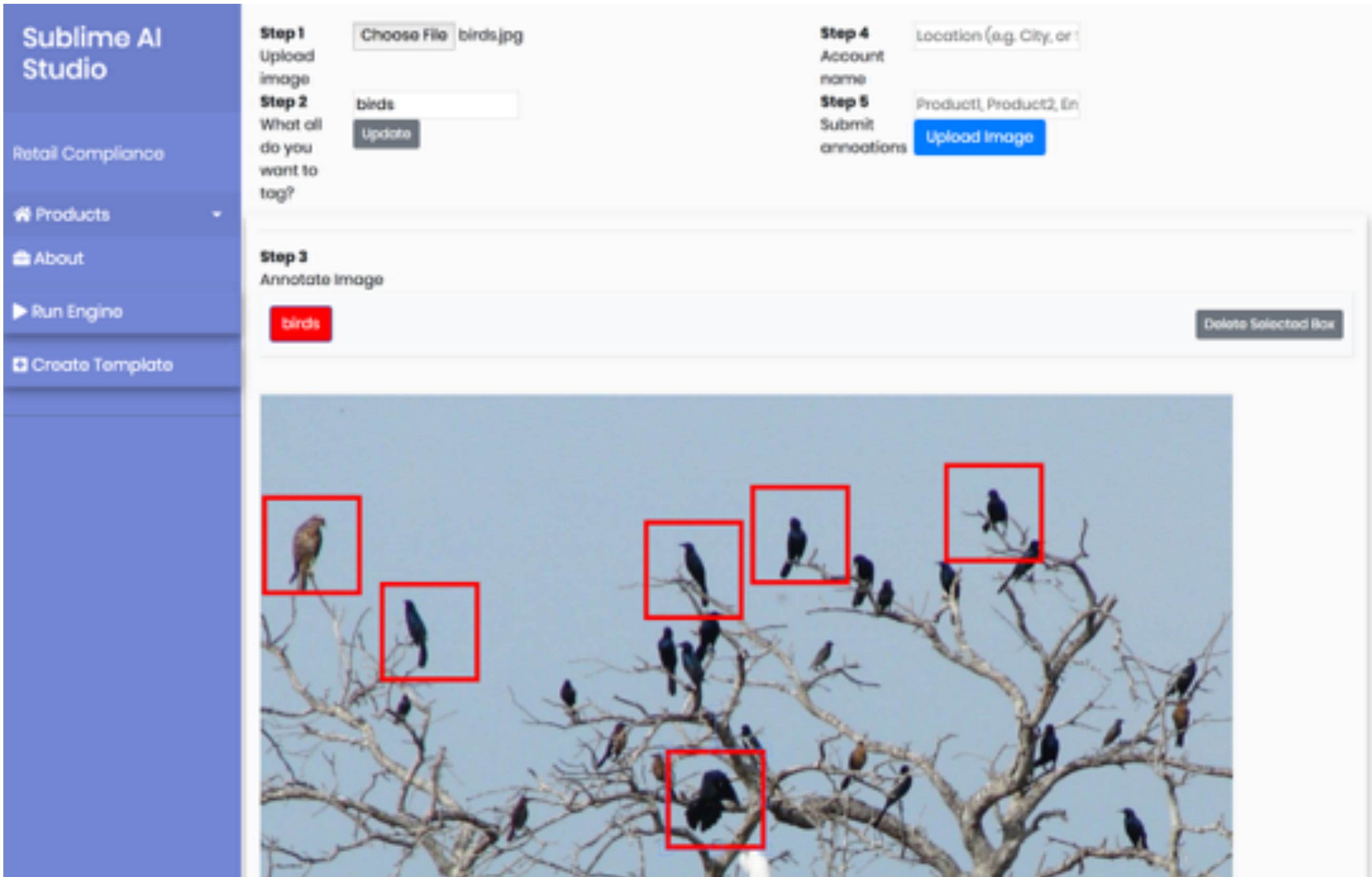
Accuracy: The process of improving the accuracy of an AI system is very tedious. For instance if an AI application for classifying dogs vs cats fails most when multiple animals are in the picture then getting more training images is hard, slow and sometimes not even practical. Resulting in often failure of AI products. TDS can enable active learning - the approach where an AI system can highlight areas in which it’s weak at, and now with the help of TDs and an AI scientist, the AI system can get a lot of data associated to that scenario. So for the example above creating more images with multiple animals is simply writing a program using TDS.

Bias in data is increasingly becoming a problem for AI systems. How do we deal with scenarios that we can’t collect? What about the case of a person jumping infant of the vehicle? Training data synthesis (TDS) is an effective way to reduce bias and programmatically create new scenarios, since we can ensure with randomization that all scenarios are equally likely to occur. Some applications want the system to learn bias, in that case as well TDS can let bias creep in by controlling the randomization parameter.

How TDS Works?

TDS using many proprietary techniques (ranging from GAN networks, Semantic Segmentation, 3D Unity Modeling). Given the requirements of the training data our design engineers use the various internal tools to generate a program that you can now call on demand to create the amount and type of training data you want for your application. To best understand TDS lets go through an example (Bird Species Identification in Parks)

1. Define a Scene



Step 1 Upload image: Choose File birds.jpg

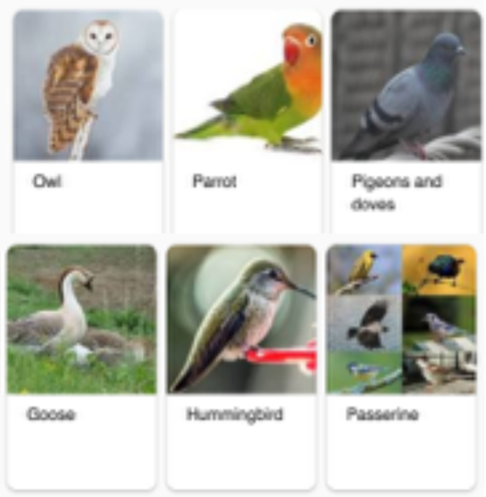
Step 2 What all do you want to tag?: birds

Step 4 Account name: Location (e.g. City, or

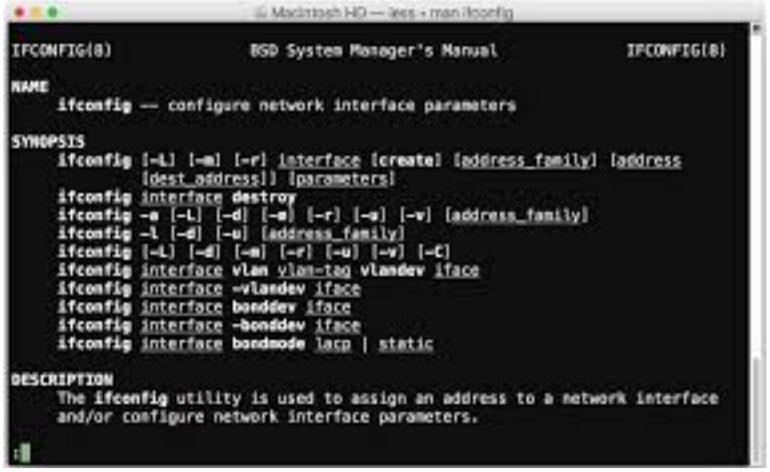
Step 5 Submit annotations: Product1, Product2, En

Step 3 Annotate image

2. Specify Bird Variations

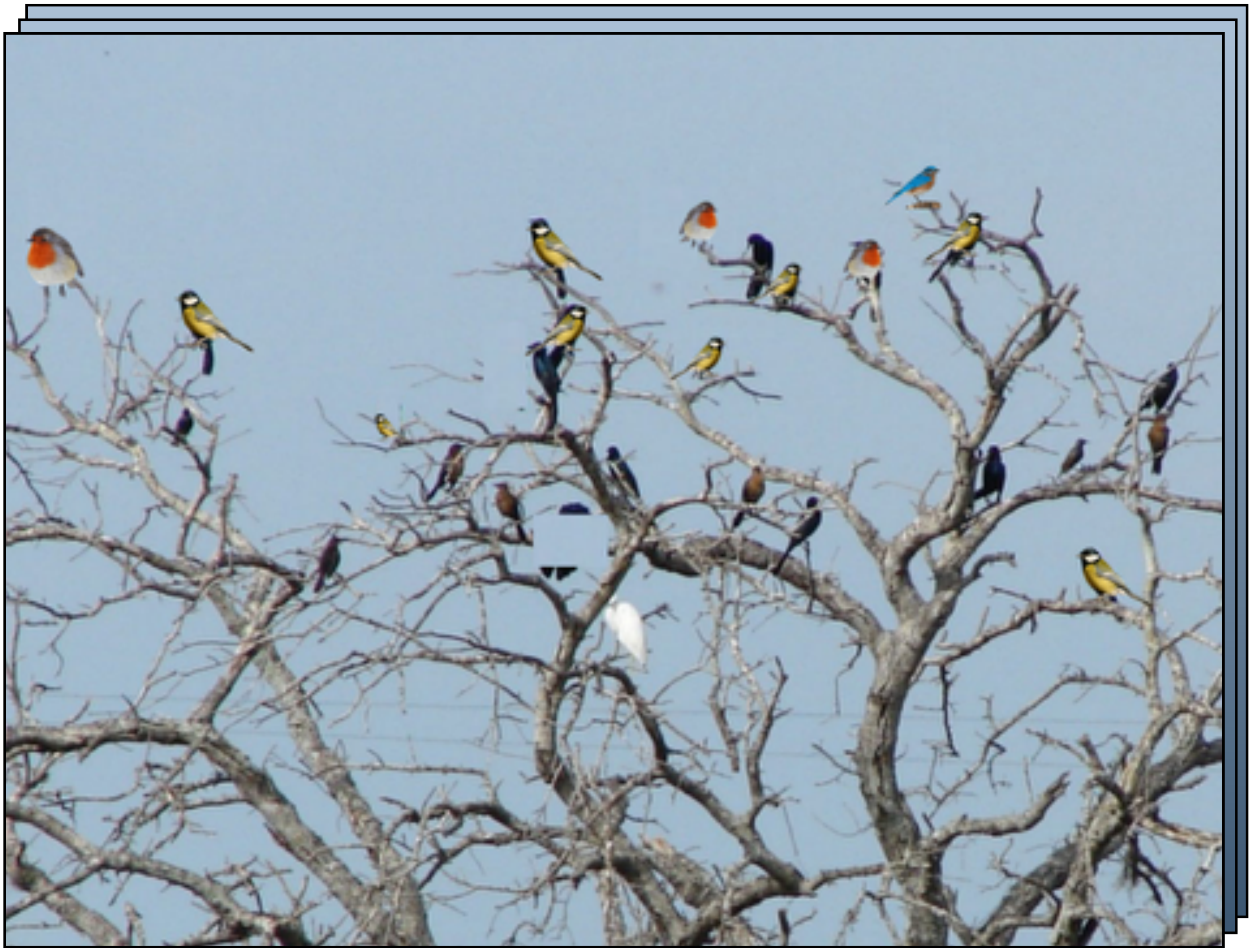


3. Specify Parameters (number of images)



```
IFCONFIG(8)                BSD System Manager's Manual               IFCONFIG(8)

NAME
     ifconfig -- configure network interface parameters

SYNOPSIS
     ifconfig [-L] [-m] [-r] interface [create] [address_family] [address
              [dest_address]] [parameters]
     ifconfig interface destroy
     ifconfig -a [-L] [-d] [-m] [-r] [-u] [-v] [address_family]
     ifconfig -l [-d] [-u] [address_family]
     ifconfig [-L] [-d] [-m] [-r] [-u] [-v] [-C]
     ifconfig interface vlan vlan-tag vlandev iface
     ifconfig interface -vlandev iface
     ifconfig interface bonddev iface
     ifconfig interface -bonddev iface
     ifconfig interface bondmode lacp | static

DESCRIPTION
     The ifconfig utility is used to assign an address to a network interface
     and/or configure network interface parameters.

:
```

4. hundreds of training images

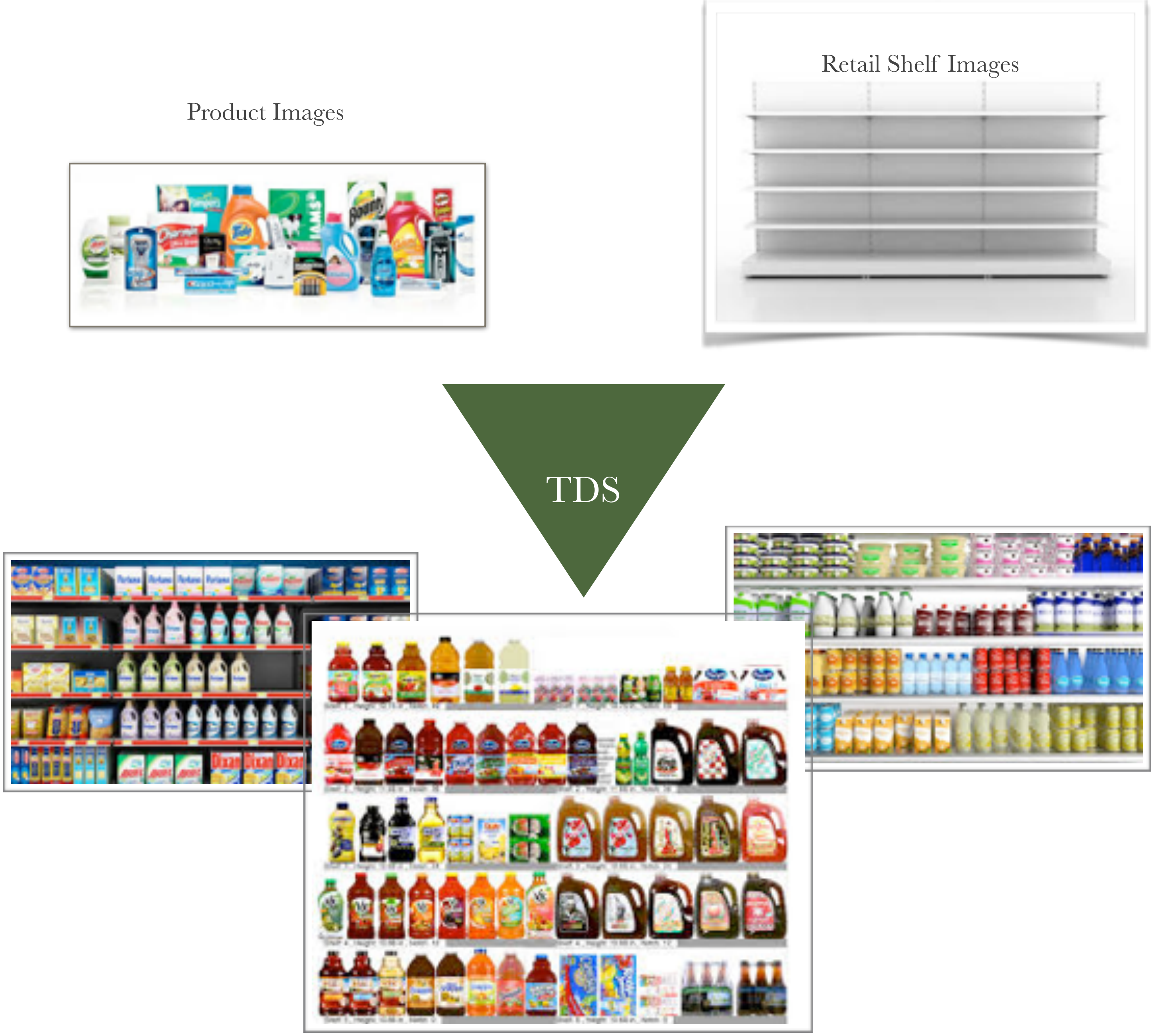


The output training images are fully customizable by code

- If you want to change the number of birds in the scene, it is a matter of writing some code using the TDS library.
- If you want to ensure only one species of birds per scene that can also be enforced with code.
- If you want to change the trees and introduce a different backgrounds that can also be done.
- If you want to ensure that the background of the birds is removed when attaching to the image that can also be programmed using the TDS library

Depending on your specific requirements we can modify the TDS code for your application.

Other example (Retail Audits)



Again, the output training images are fully customizable:

- If you want to produce images with different grid layouts that can be programmed
- If you want to produce images of products on different shelves that can be programmed as well
- If you want images with different angles then TDS engineers can create a 3D model and take snapshots of different views and illuminations.

Please note: TDS libraries are not available for direct use by end users. Synthetic training data are created using a variety of proprietary methods so it requires the use of tools as well as engineers to accomplish the desired training images you want.

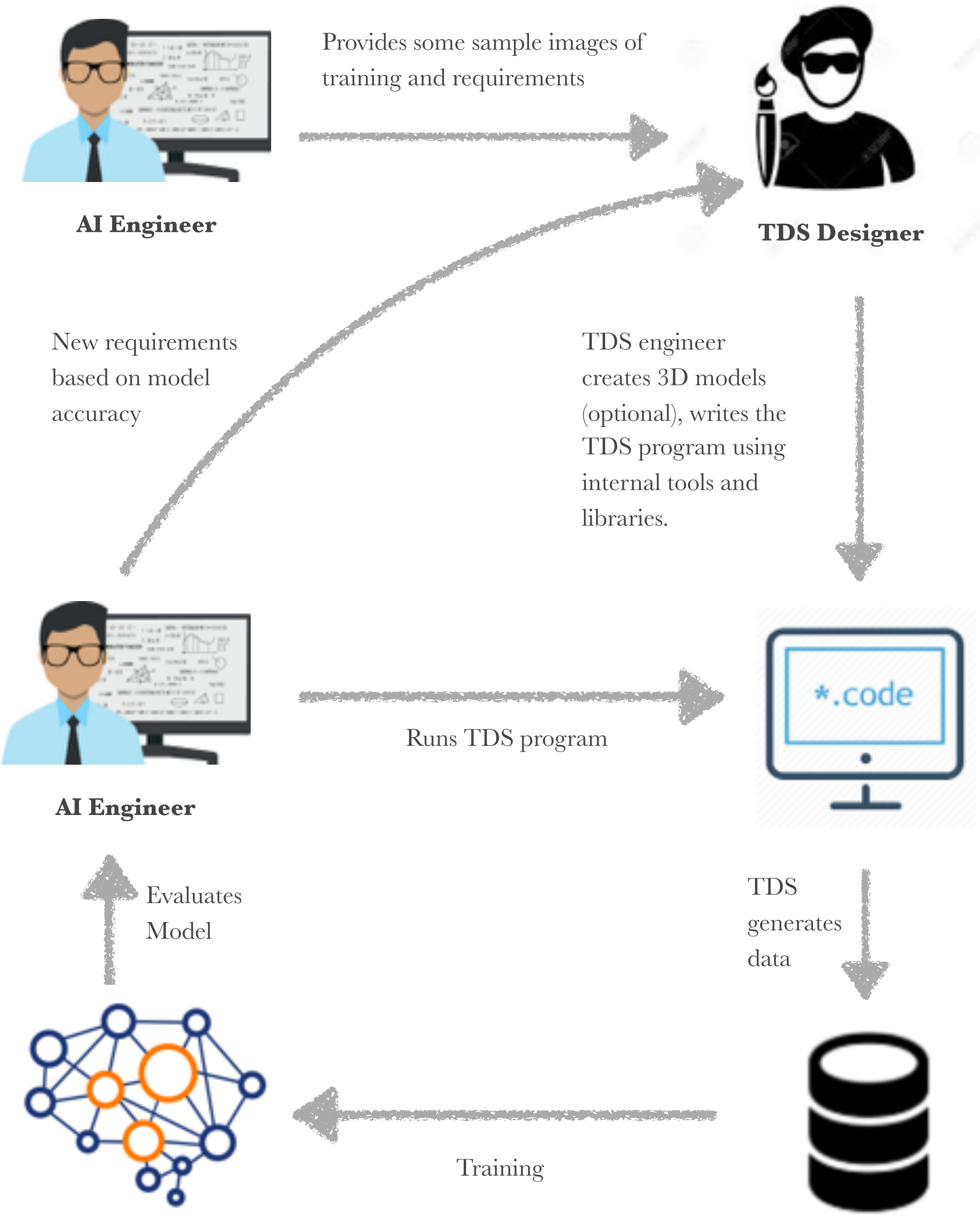
How does TDS work with human labelled data?

TDS integrates beautifully with human labelled data. Rather the recommended approach is to use TDS for train and test phase but use human labeled data for validation. The benefit is that synthetic data will force the system to learn to learn generalizations and the human labeled data will act as a gate keeper to ensure that the AI system will work in the field for the intended task. It also makes sense from a scaling point of view since a lot less validation data is needed compared to training data.

Since TDS turns training data into a programming paradigm it is possible to test and iterate on lower end developer GPU systems so that bugs, memory leaks, etc can be fixed before throwing the big production grade data to the systems.

How TDS is used is of course to the creativity of the AI scientist, like any powerful tech, it requires careful consideration. We have seen companies mix TDS with human labeled data. For instance they take a picture of the scene in which they want to classify, then use TDS to create many variations.

TDS Workflow



Provides some sample images of training and requirements

AI Engineer

TDS Designer

New requirements based on model accuracy

TDS engineer creates 3D models (optional), writes the TDS program using internal tools and libraries.

AI Engineer

Runs TDS program

Evaluates Model

TDS generates data

Training