ature is in a pre-release state and might change or have limited support. For more information, see the product launc
(/products/#product-launch-stages).

This page provides detailed reference information about arguments you submit to AI Platform when running a training job using the built-in image classification algorithm.

The built-in image classification algorithm uses TensorFlow 1.14.

Arguments	Details
training_data_path	Path to a **TFRecord** path pattern used for training. It can be in glob pattern, e.g., • gs://input_data_bucket/train-001-100.tfrec • gs://input_data_bucket/train* • gs://input_data_bucket/train-0??-100.tfrec • gs://input_data_bucket/train-00[1-9]-100.tfrec **Required** **Type:** String

Arguments	Details
validation_data_path	Path to a **TFRecord** path pattern used for validation. It can be in glob pattern, e.g., • gs://input_data_bucket/validation-001-100.tfrec • gs://input_data_bucket/validation* • gs://input_data_bucket/validation-0??-100.tfrec • gs://input_data_bucket/validation-00[1-9]-100.tfrec **Required** **Type:** String
job-dir	Path where model, checkpoints and other training artifacts will reside. The following directories will be created here: • model: It will contain the **best** trained SavedModel • checkpoints **Required** **Type:** String

The built-in image classification algorithm has the following hyperparameters:

Hyperparameter	Details
BASIC PARAMETERS	
num_classes	The number of classes in the training/validation data. It should match the classes in training/validation dataset. E.g., for **`num_classes`**=5, the range of **`image/class/label`** in input tf.Example needs to be [0, 4]. **Required** **Type:** Integer
max_steps	The number of steps that the training job will run. After **`max_steps`**, the training job will finish automatically. **Required** **Type:** Integer

Hyperparameter	Details
train_batch_size	The number of images used in one training step. If this number is too big, the job may fail with out-of-memory (OOM). **Default:** 32 **Type:** Integer
num_eval_images	The number of total images used for evaluation. Its value needs to be equal or less than the total images in the `validation_data_path`. **Default:** 0 **Type:** Integer
pretrained_checkpoint_path	The path to pretrained checkpoints. A good pre-trained checkpoint would be helpful to increase the model convergence speed or achieve better model quality. See `gs://builtin-algorithm-data-public/pretrained_checkpoints/classification/` for some pretrained checkpoints.
Learning Rate Parameters	
learning_rate_decay_type	The method by which the learning rate decays during training. **Default:** '*cosine*' **Type:** String **Options:** one of *{cosine, stepwise}*
warmup_learning_rate	The initial learning rate during warm-up phase. **Default:** 0 **Type:** Float
warmup_steps	The number of steps to run during the warm-up phase, or the length of the warm-up phase in steps. The training job uses **`warmup_learning_rate`** during the warm-up phase. When the warm-up phase is over, the training job uses **`initial_learning_rate`**. **Default:** 0 **Type:** Integer
initial_learning_rate	The initial learning rate after warmup period. **Default:** 0.0001 **Type:** Float

Hyperparameter	Details
stepwise_learning_rate_steps	The steps to decay/change learning rates for **`stepwise`** learning rate decay type. For example, **`100,200`** means the learning rate will change (with respect to stepwise_learning_rate_levels) at step 100 and step 200. Note that it will be respected only when **`learning_rate_decay_type`** is set to **`stepwise`**. **Default:** 100,200 **Type:** String
stepwise_learning_rate_levels	The learning rate value of each step for **`stepwise`** learning rate decay type. Note that it will be respected only when **`learning_rate_decay_type`** is set to **`stepwise`**. **Default:** 0.008,0.0008 **Type:** String
Optimizer Parameters	
optimizer_type	The optimizer used for training. **Default:** 'momentum' **Type:** String **Options:** one of *{momentum, adam, rmsprop}*
optimizer_arguments	The arguments for optimizer. It is a comma separated list of "name=value" pairs. It needs to be compatible with **`optimizer_type`**. For example, • For Momentum optimizer, it accepts "momentum=0.9". See tf.train.MomentumOptimizer (https://www.tensorflow.org/api_docs/python/tf/train/MomentumOptimizer) for more details. • For Adam optimizer, it can be "beta1=0.9,beta2=0.999". See tf.train.AdamOptimizer (https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer) for more details. • For RMSProp optimizer, it can be "decay=0.9,momentum=0.1,epsilon=1e-10". See RMSPropOptimizer (https://www.tensorflow.org/api_docs/python/tf/train/RMSPropOptimizer) for more details. **Type:** String
Model parameters	

Hyperparameter	Details
image_size	The image size (width and height) used for training. Note that the training job may be OOM if its value is too big. **Default:** 224 **Type:** Integer
model_type	That model architecture type used to train models. **Type:** String **Options:** one of {resnet-18, resnet-34, resnet-50, resnet-101, resnet-152, resnet-200, efficientnet-b0, efficientnet-b1, efficientnet-b2,, efficientnet-b3, efficientnet-b4, efficientnet-b5, efficientnet-b6, efficientnet-b7}
label_smoothing	Label smoothing parameter used in the softmax_cross_entropy. **Default:** 0 **Type:** Float **Options:** [0,1)
weight_decay	Weight decay co-efficiant for l2 regularization, e.g., `loss = cross_entropy + params['weight_decay']*l2_loss`. **Default:** 0.0001 **Type:** Float **Options:** [0,+∞)

Hyperparameter tuning tests different hyperparameter configurations when training your model. It finds hyperparameter values that are optimal for the selected goal metric. For each tunable argument, you can specify a range of values to restrict and focus the possibilities AI Platform can try.

Learn more about hyperparameter tuning on AI Platform (/ml-engine/docs/hyperparameter-tuning-overview).

For the image classification algorithm, the only option is to maximize accuracy:

Objective Metric	Direction	Details

Objective Metric	Direction	Details
top_1_accuracy	MAXIMIZE	The highest prediction accuracy.

When training with the built-in image classification algorithm, you can tune the following hyperparameters. Start by tuning parameters with "high tunable value." These have the greatest impact on your goal metric.

Hyperparameters	Type	Range/Values
PARAMETERS WITH HIGH TUNABLE VALUE		
initial_learning_rate		[0, inf]
learning_rate_decay_type		one of {cosine, stepwise}
max_steps		[0, inf]
optimizer_type		one of {momentum, adam, rmsprop}
OTHER PARAMETERS		

Hyperparameters	Type	Range/Values
optimizer_arguments		See hyperparameter section for more details.
image_size		
model_type		See hyperparameter section for more details.
label_smoothing		[0, 1)
weight_decay		[0, inf]