Learning multiple visual domains with residual adapters
Early Transfer learning
* Use activation of fc layers as feature extractor
* Use ANN or other ML technique on fc features
Background

Fine tuning
* New domain has large dataset → you may tune all parameters in network

Causes catastrophic forgetting:
* Network is no longer good at old problem
Background

Catastrophic forgetting:
* I think it is a non issue
* If training data is too much to store other methods could be used
* For example reservoir sampling
* Could be useful in online learning
Joint training:
- Have multiple datasets
- Train on all datasets at the same time
- Does not cause catastrophic forgetting
Background

Less forgetting learning
* Try to not change fc output too much when doing transfer learning to avoid CF
Background

Learning without forgetting
* Tune fc layers for old tasks to make output similar to before
Background

Different batch norm for different domains
Add new weights for new domains, only train new weights.
* Train all new weights on new domain
* since old weights not trained $\rightarrow$ no CF
Background

ResNet

Conv+BN+ReLU → Conv+BN+ReLU +
New architecture

Replace Conv+BN+RELU with

Where BN has domain specific parameters
New architecture

Possible to tune network on a per domain basis
“Visual decathlon challenge”
- Test performance of multi domain learning on 10 domains
- only classification tasks
New evaluation method

- Evaluation metric seems strange
  “we double the error rates of the fully finetuned networks”
- i.e. for max score you need half the error of the baseline (?)
Finding domain

- Some part about classifying which domain the input is from

- Often input domain should be known from, for example metadata.
On “finetune all” all domains were finetuned jointly, which explains the #par for model

"network accept 64x64 images as input" → all input scaled to this size (?)
Summary

* Introduce way to add domain specific parameters
* Training on all image categories (Res. Adapt finetune all) gave superior results to domain specific fine tuning.
* Benchmark for multiple domains