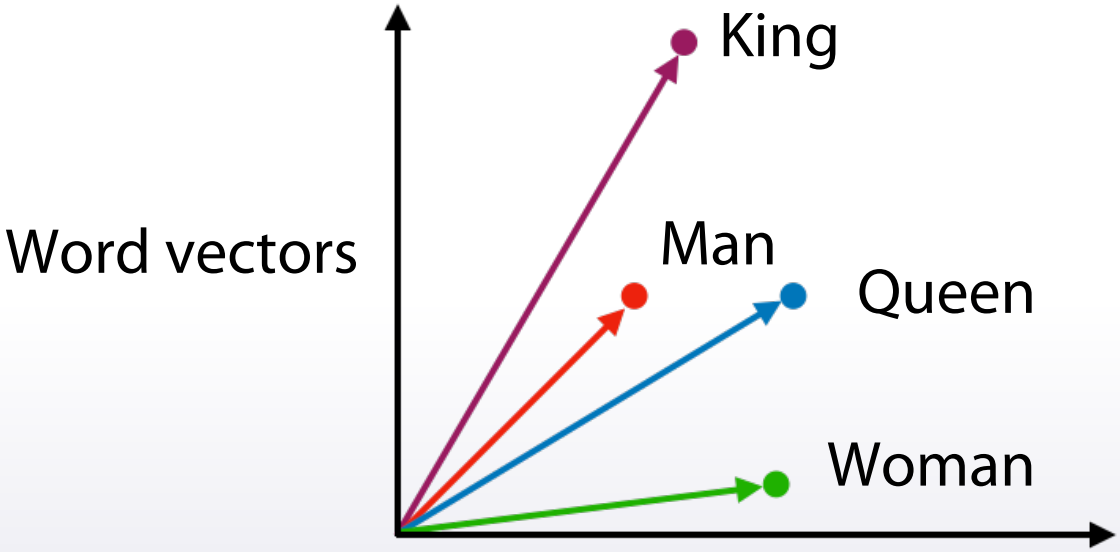
Text -> vector

1. Bag of words:

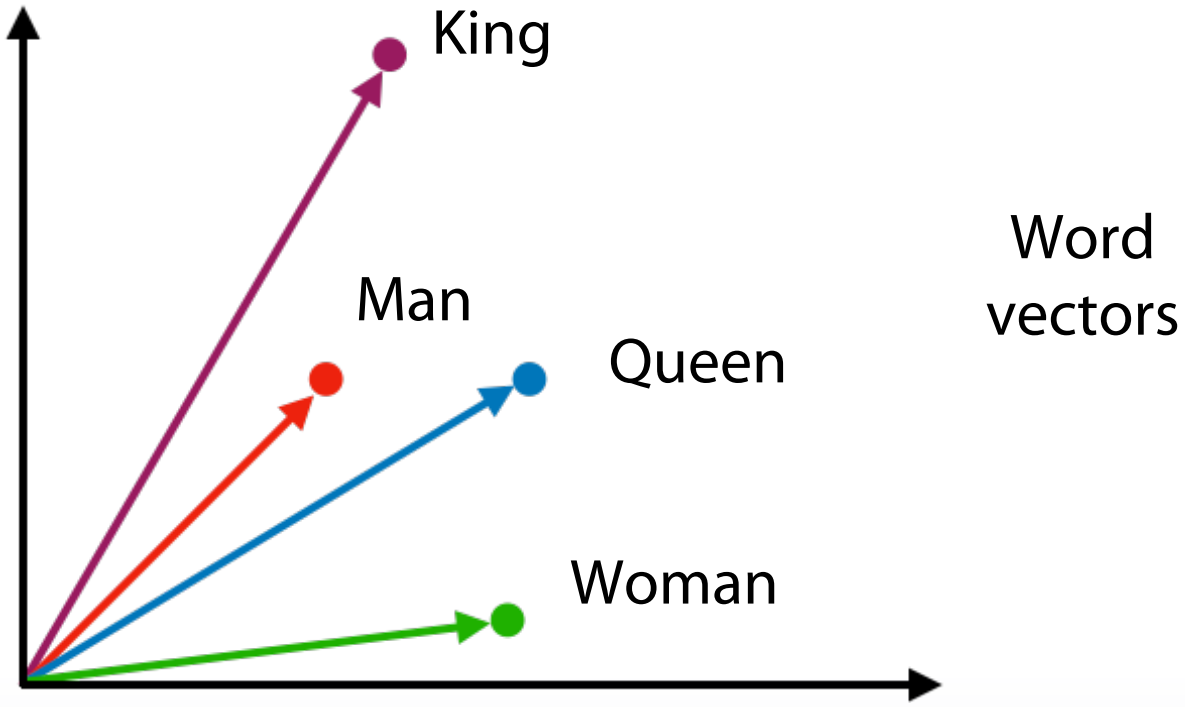
The dog is on the table

0	0	1	1	0	1	1	1
are	cat	dog	is	now	on	table	the

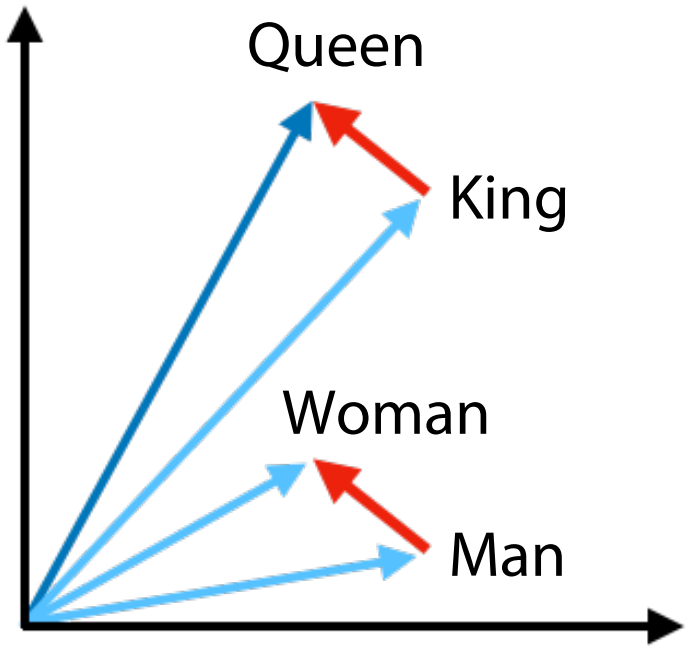
2. Embeddings (~word2vec):



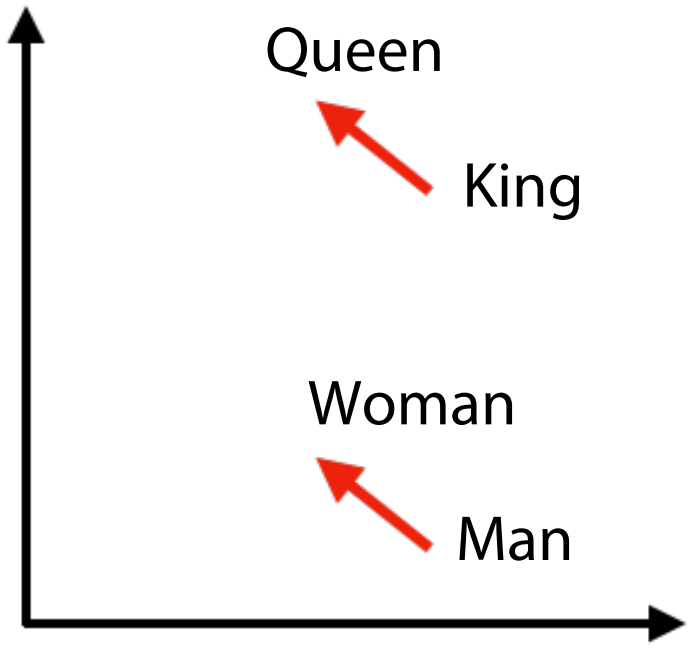
Word2vec



Word2vec



king + woman - man = queen



The red direction encodes gender

Word2vec

stands for Global Vector for word representation.

Words: Word2vec, Glove, FastText, etc

Sentences: Doc2vec, etc

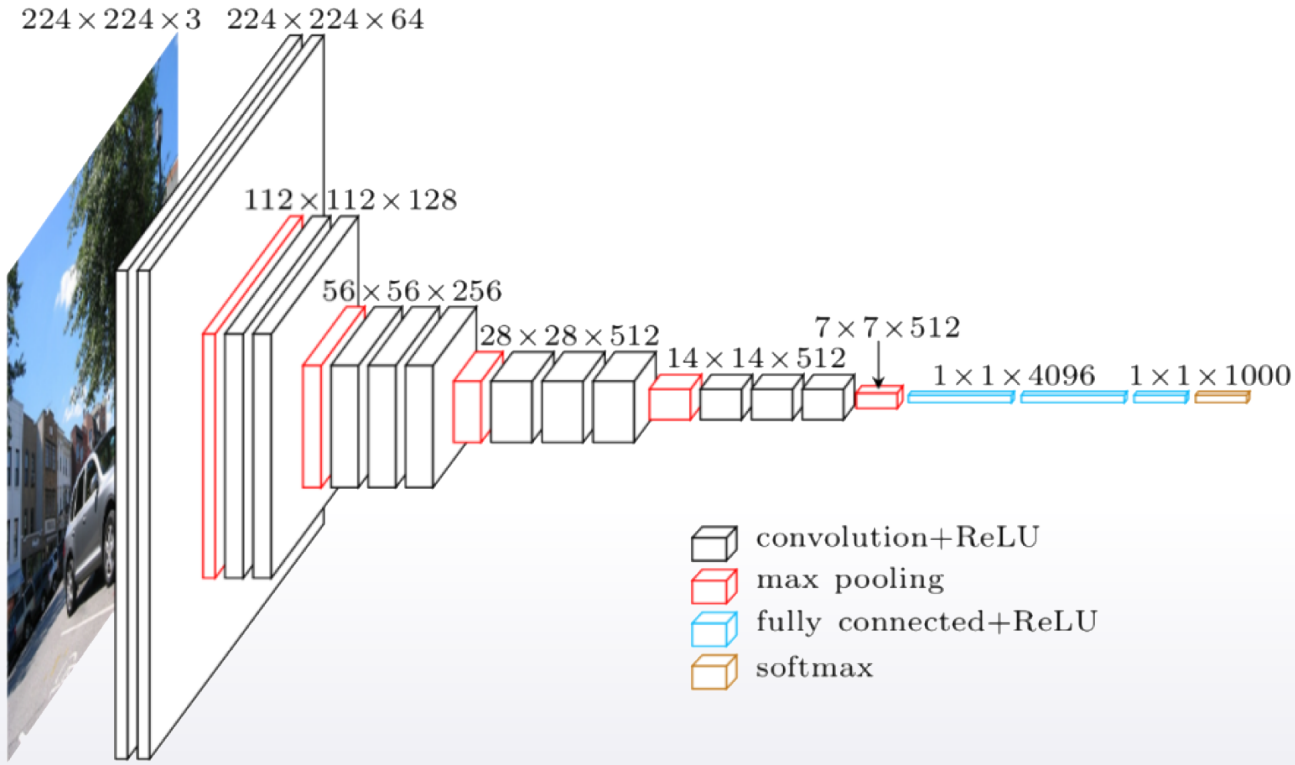
There are pretrained models

BOW and w2v comparison

1. Bag of words
    a. Very large vectors
    b. Meaning of each value in vector is known
2. Word2vec
    a. Relatively small vectors
    b. Values in vector can be interpreted only in some cases
    c. The words with similar meaning often have similar embeddings

Image -> vector

1. Descriptors
2. Train network from scratch
3. Finetuning



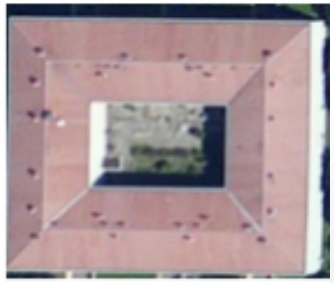
Finetuning example

Category 1:
North-South orientation

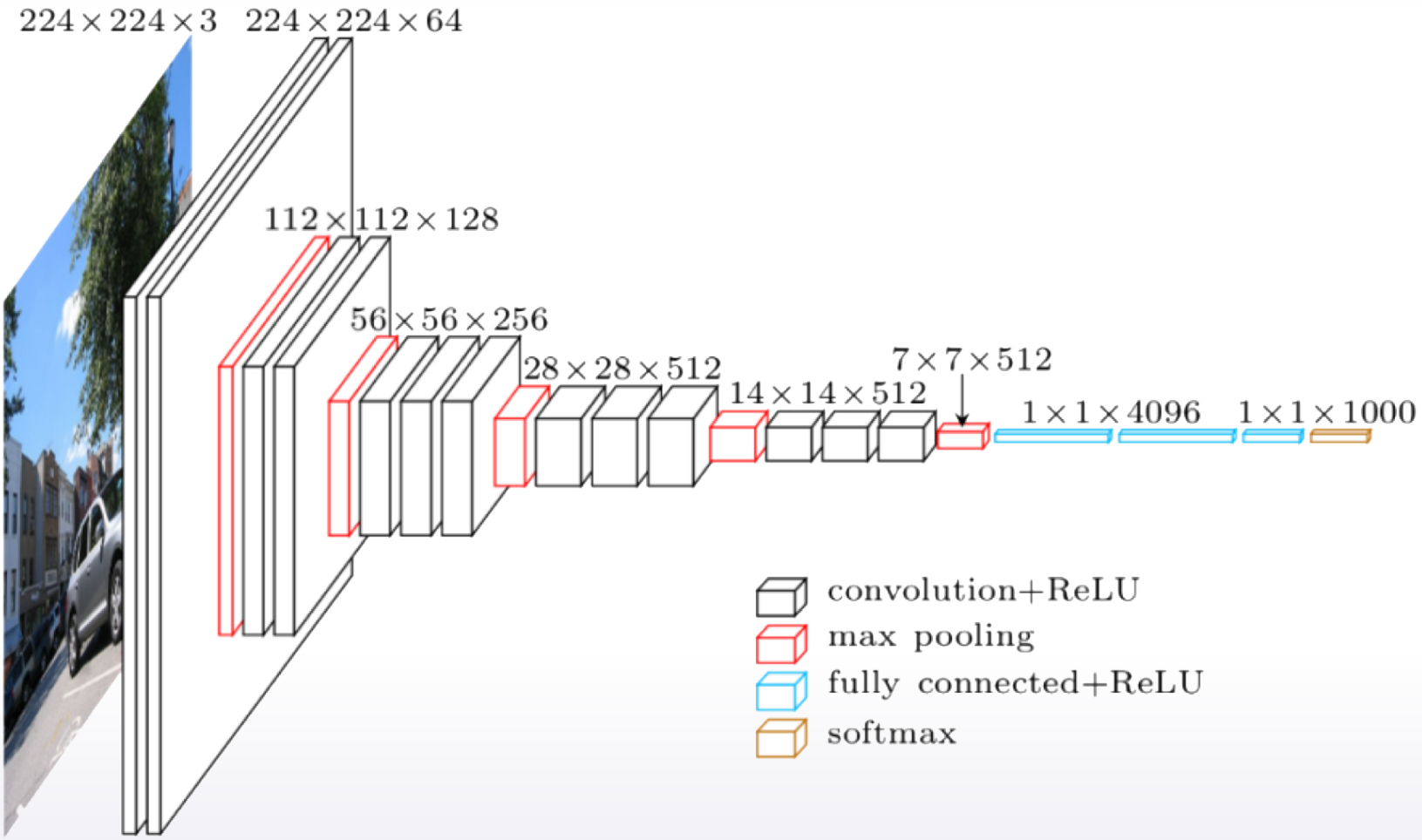
Category 2:
East-West orientation

Category 3:
Flat roof

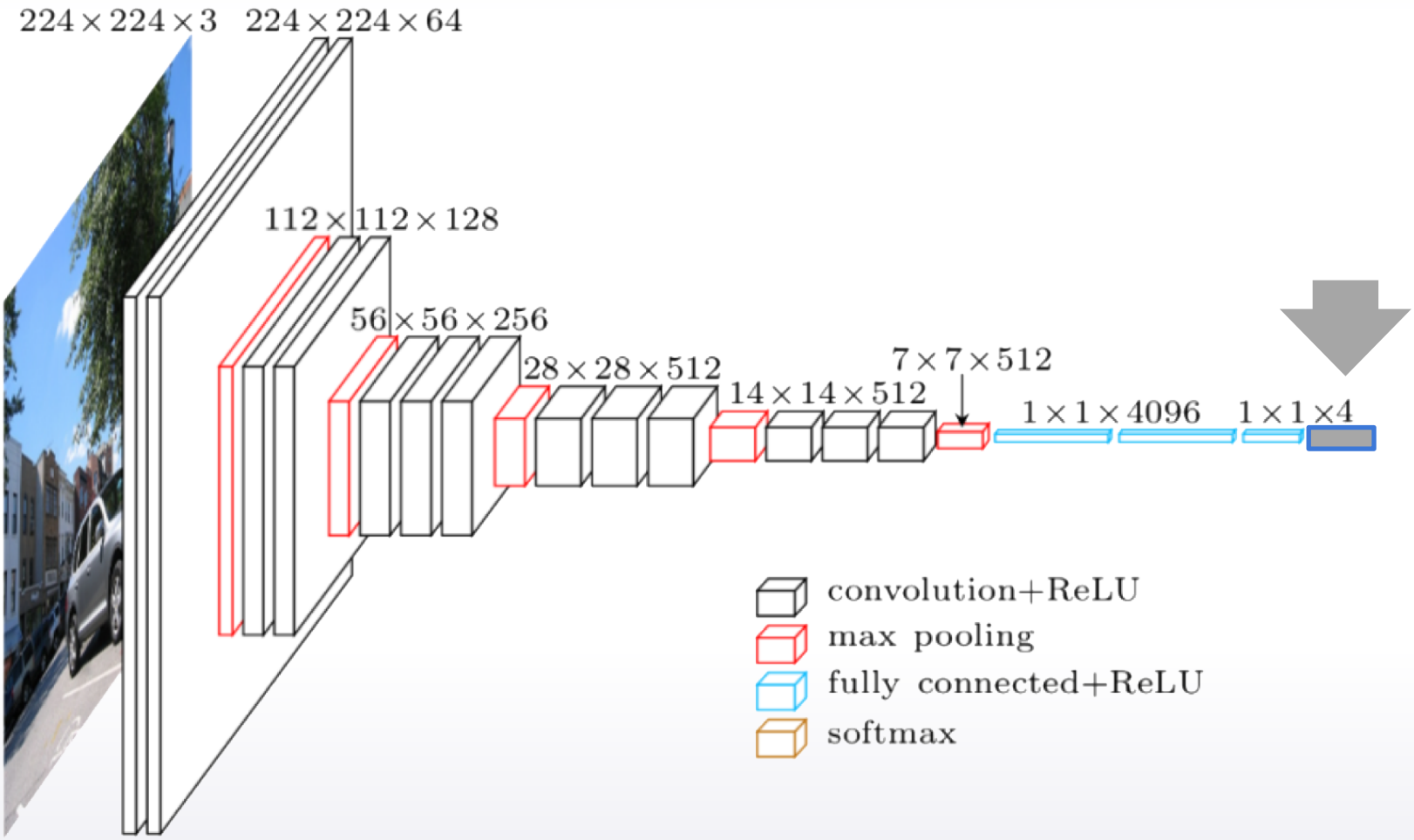
Category 4:
Other



Finetuning example



Finetuning example



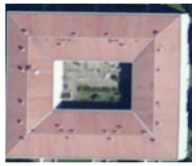
Augmentation

Category 1:
North-South orientation

Category 3:
Flat roof

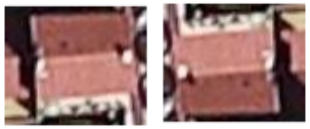
Category 2:
East-West orientation

Category 4:
Other



Augmentation

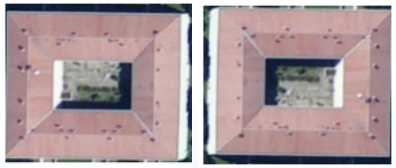
Category 1:
North-South orientation



Category 3:
Flat roof

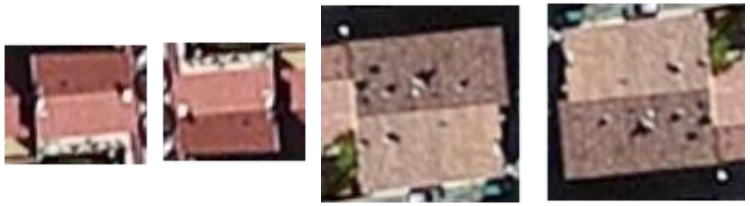
Category 2:
East-West orientation

Category 4:
Other



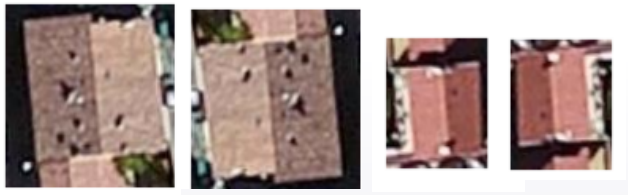
Augmentation

Category 1:
North-South orientation

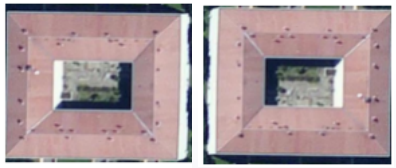


Category 3:
Flat roof

Category 2:
East-West orientation

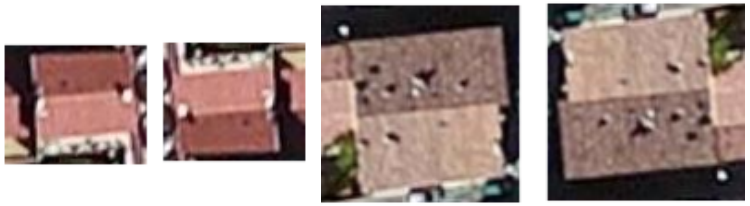


Category 4:
Other



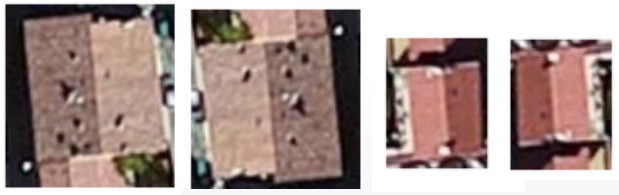
Augmentation

Category 1:
North-South orientation

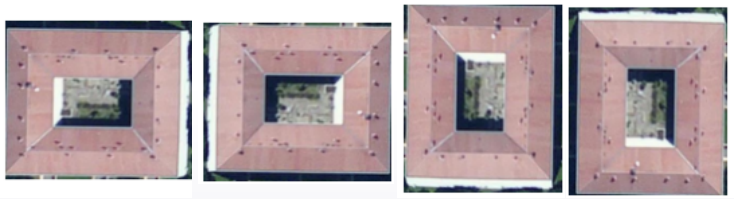


Category 3:
Flat roof

Category 2:
East-West orientation



Category 4:
Other



Feature extraction from text and images

1. **Texts**
    a. Preprocessing
        i. Lowercase, stemming, lemmarization, stopwords
    b. Bag of words
        i. Huge vectors
        ii. Ngrams can help to use local context
        iii. TFiDF can be of use as postprocessing
    c. Word2vec
        i. Relatively small vectors
        ii. Pretrained models

Feature extraction from text and images

1. **Texts**
    a. Preprocessing
        i. Lowercase, stemming, lemmarization, stopwords
    b. Bag of words
        i. Huge vectors
        ii. Ngrams can help to use local context
        iii. TFiDF can be of use as postprocessing
    c. Word2vec
        i. Relatively small vectors
        ii. Pretrained models
2. **Images**
    a. Features can be extracted from different layers
    b. Careful choosing of pretrained network can help
    c. Finetuning allows to refine pretrained models
    d. Data augmentation can improve the model