Visual Intelligence Platform

Deep Video Analytics + Visual Data Network

Akshay Bhat
Cornell Tech, Cornell University.

An overview of computer vision research by Tomasz Malisiewicz

http://www.computervisionblog.com/2015/01/from-feature-descriptors-to-deep.html

Quick summary

Sift, Graph Cuts

HOG, DPM

Deep Learning

?

Caltech 101, Matlab, OpenCV

VOC, Imagenet, Caffe, Theano

?

Numerous high quality libraries & datasets

- OpenCV
- ROS
- Caffe (model zoo!)
- Theano
- Torch
- Tensor Flow
- CNTK
- MXNET
- Pytorch

- Caltech 101
- Imagenet
- COCO
- Too many to keep track!
  - Open Images
  - Soundnet
  - Mapnet
  - CMU Video patch dataset

A deluge of datasets!

- VideoNet
- Yahoo Flickr Creative Commons 100M
- ViCom
- Visual Genome
- YouTube-BoundingBoxes
- Youtube 8M
- imSitu by AllenAI
- Charades by Allen AI
- Udacity car dataset
- KITTI
- Caltech, INRIA, ETH Pedestrians
- Stanford Drone Dataset
- COCO text

We are reaching a stage where

Number of datasets ≅ Number of research groups

With each dataset having its own JSON or XML format, incompatible with all others.

State of the art pre trained models

- Imagenet classification
  - Inception
  - Resnet
  - VGG
- Detection models
  - R-CNN
  - YOLO
  - SSD
- Face detection / recognition
  - Face-MTCNN
  - Facenet
- Semantic Segmentation models
  - Multipathnet
  - FCN
- Audio embedding models
  - Soundnet

What is hidden in plain sight?

We need a platform which seamlessly combines

Data + Models + User Interface

A Relational Model of Data for Large Shared Data Banks. By Edgar F. Codd

Can we develop an equivalent of relational model / databases for visual data?

Visual Data

=

{ Images, Videos, Annotations, Features}

Relational data : Postgres, MYSQL, SQLite

::

Text, HTML : Lucene/Solr, Elasticsearch

::

Videos & Images : _____

Previous attempts: Lire project

- LIRE: Lucene Image Retrieval
  - http://www.lire-project.net/
- Developed pre Deep Learning
- Functionality limited to computing & storing feature vectors such as Color Layout, Edge Histogram, etc.

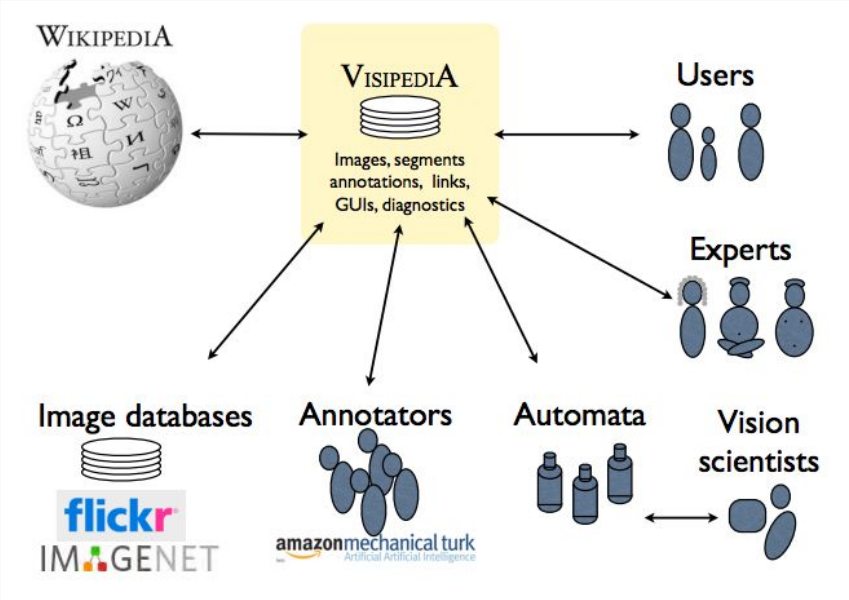
Previous attempts: CloudCV

- Large Scale Distributed Computer Vision as a Cloud Service
- Support for OpenCV, Graphlab, Cafe
- Image Classification, VQA, stitching, etc
- Does not retains state. E.g. you cannot store images.

Previous attempts: NVidia DIGITS

- ”DIGITS (the Deep Learning GPU Training System) is a webapp for training deep learning models. ”
- Load/create datasets, train models, deploy models.
- Aimed at researchers
- Written in Python/Flask with Torch & Caffe supported

Previous attempts: Visipedia



Taken from Vision of a Visipedia, Perona et. al.

Previous attempts: Visipedia

- Collaborative creation of visual data
- Pre-defined set of concepts E.g. Birds, Trees
- Different type of participants
  - Experts, Annotators, Citizen Scientists, Users, Computer scientists
- Retains state

Previous attempts: VMX.ai

- Underfunded Kickstarter project Circa Jan 2014
- by Tomasz Malisiewicz
- Pre Tensor Flow, Pre Deep Learning
- Allow developers to create real time detectors
- Support for training model

Why now?

- High quality libraries and pre-trained models
  - TensorFlow
  - Inception, SSD, Facenet
  - Flickr LOPQ, Facebook FAISS
- Cheap GPUs (local & cloud)
- Docker enables deployment of complex applications

Relational data : Postgres, MYSQL, SQLite

::

Text, HTML : Lucene/Solr, Elasticsearch

::

Videos & Images : _____

Relational data : Postgres, MYSQL, SQLite

::

Text, HTML : Lucene/Solr, Elasticsearch

::

Videos & Images : ***Deep Video Analytics***

People : Facebook, MySpace

::

Code : Git / GitHub, GitLab

::

Visual Data: ***Visual Data Network***

Relational data : SQL

::

Text, HTML : inverted word index, Page Rank

::

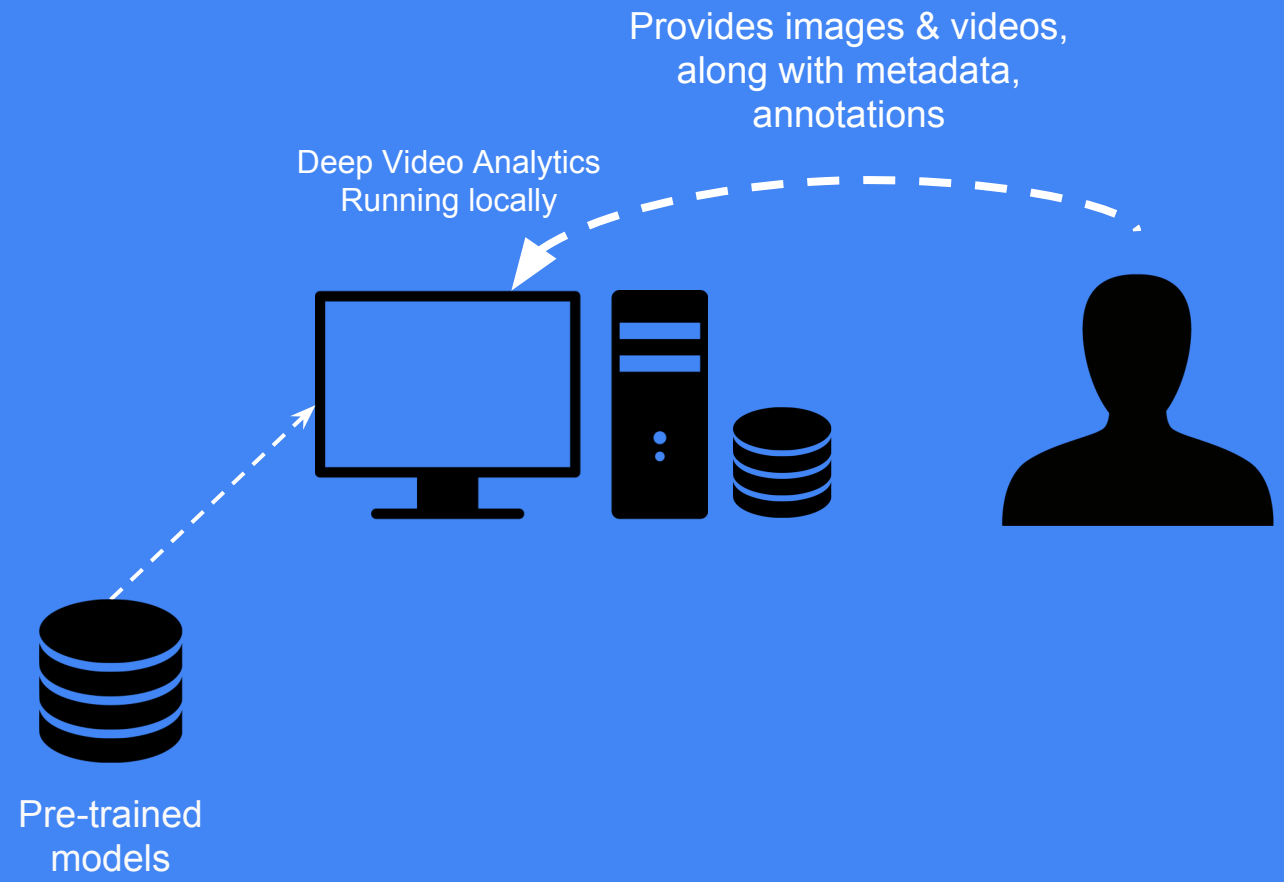
Videos & Images : ***Approximate Nearest Neighbor***

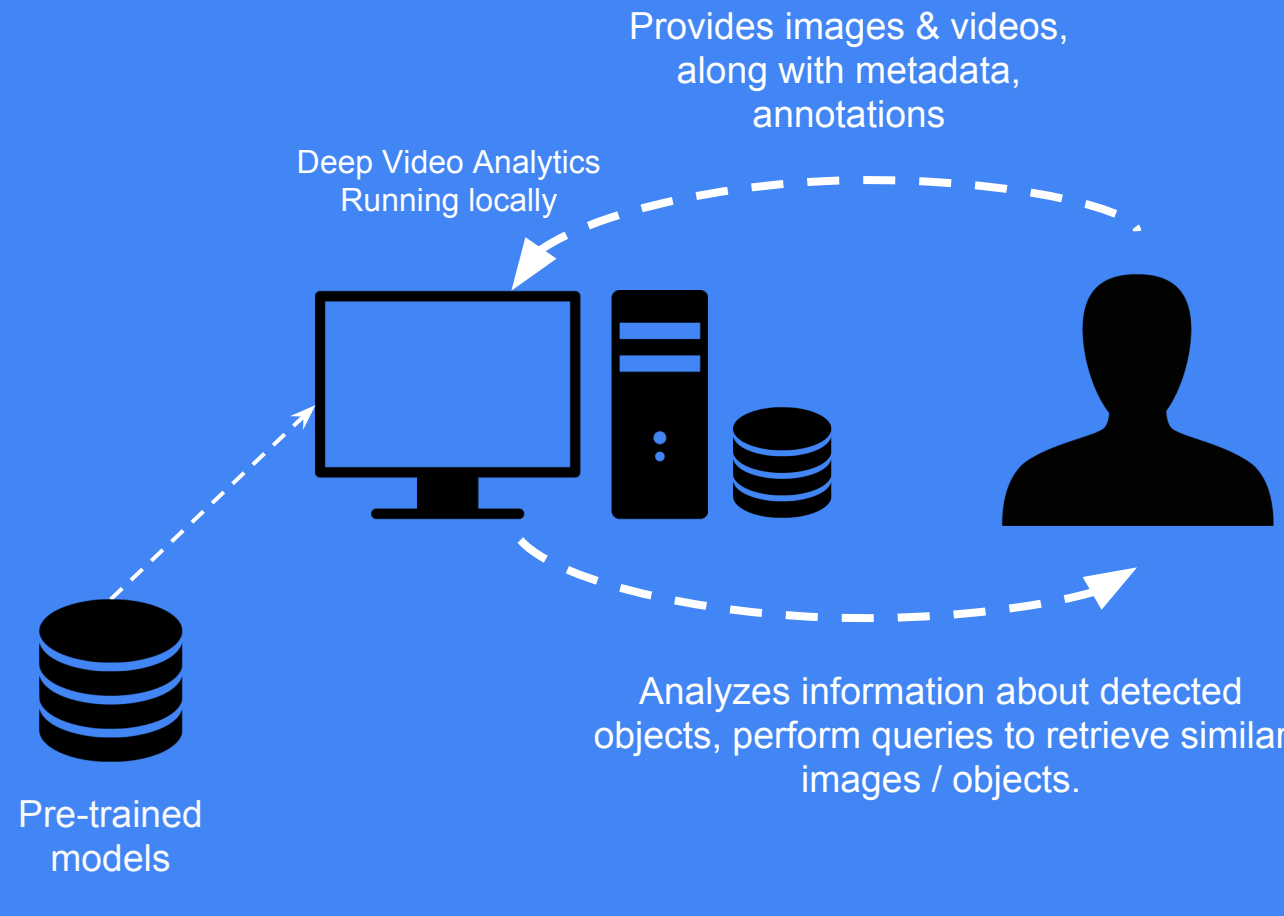
Provides images & videos,
along with metadata,
annotations

Deep Video Analytics
Running locally

Provides images & videos,
along with metadata,
annotations

Deep Video Analytics
Running locally

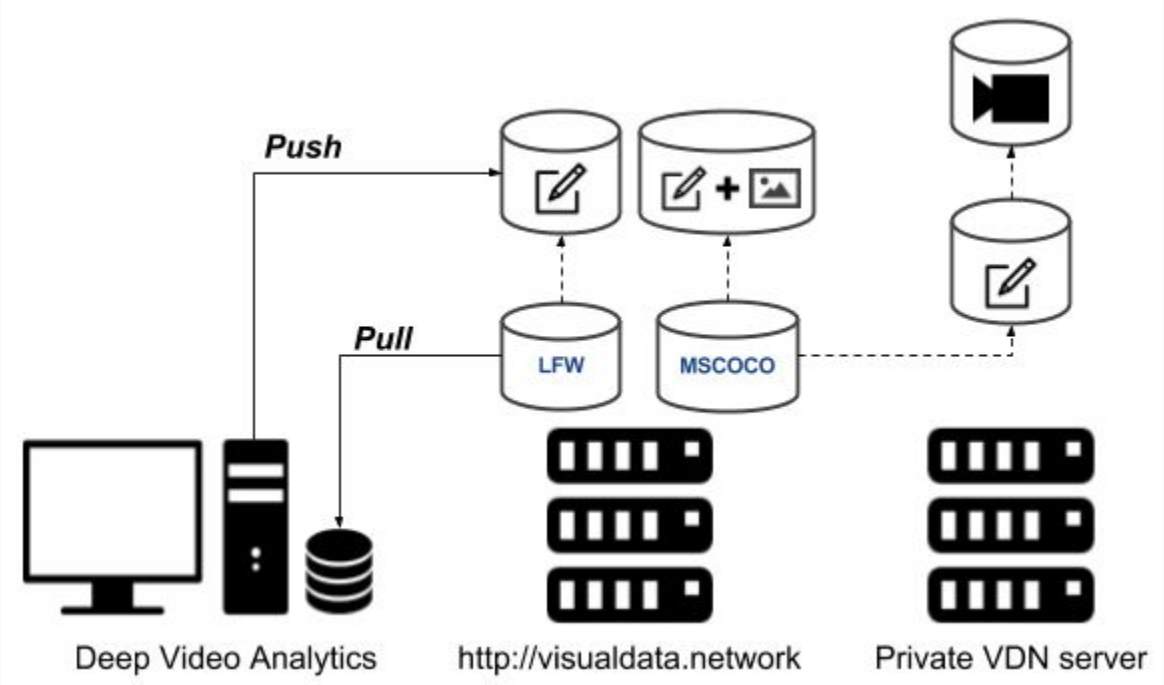

Provides images & videos,
along with metadata,
annotations
Deep Video Analytics
Running locally
Pre-trained
models


Provides images & videos,
along with metadata,
annotations
Deep Video Analytics
Running locally
Analyzes information about detected
objects, perform queries to retrieve similar
images / objects.
Pre-trained
models

Deep Video Analytics enables rapid data creation

Visual Data Network allows seamless sharing

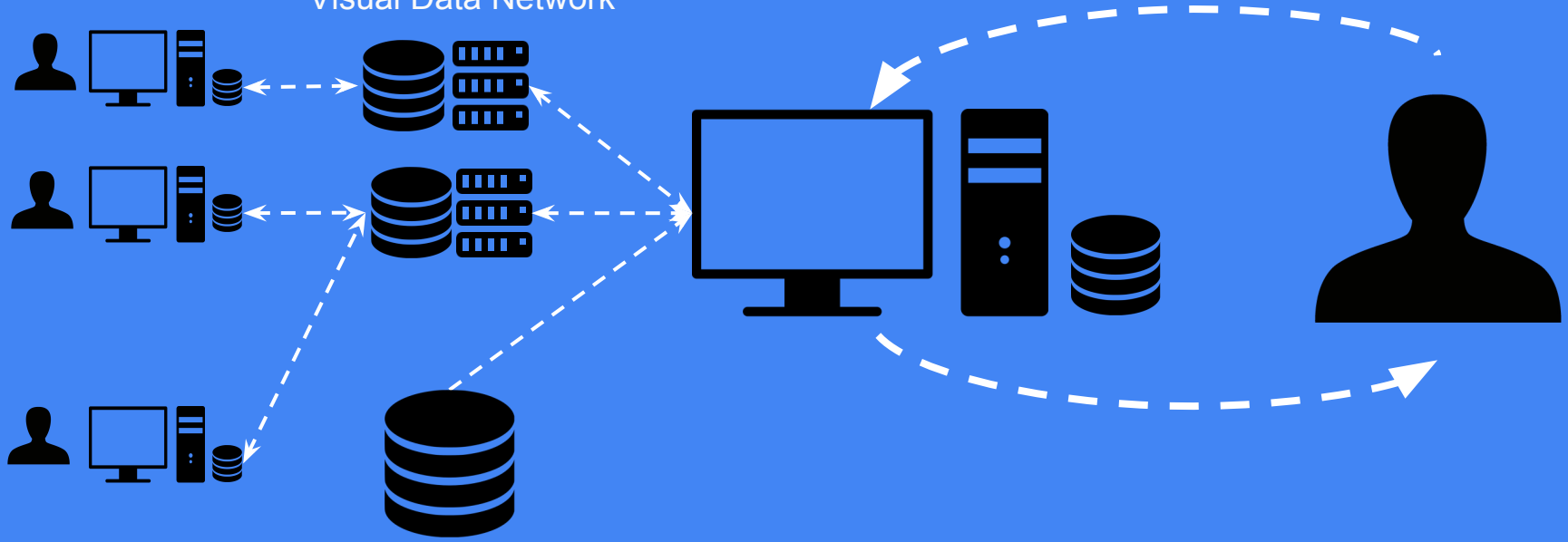
Push, Pull video / dataset, Annotations, just like you would with GitHub



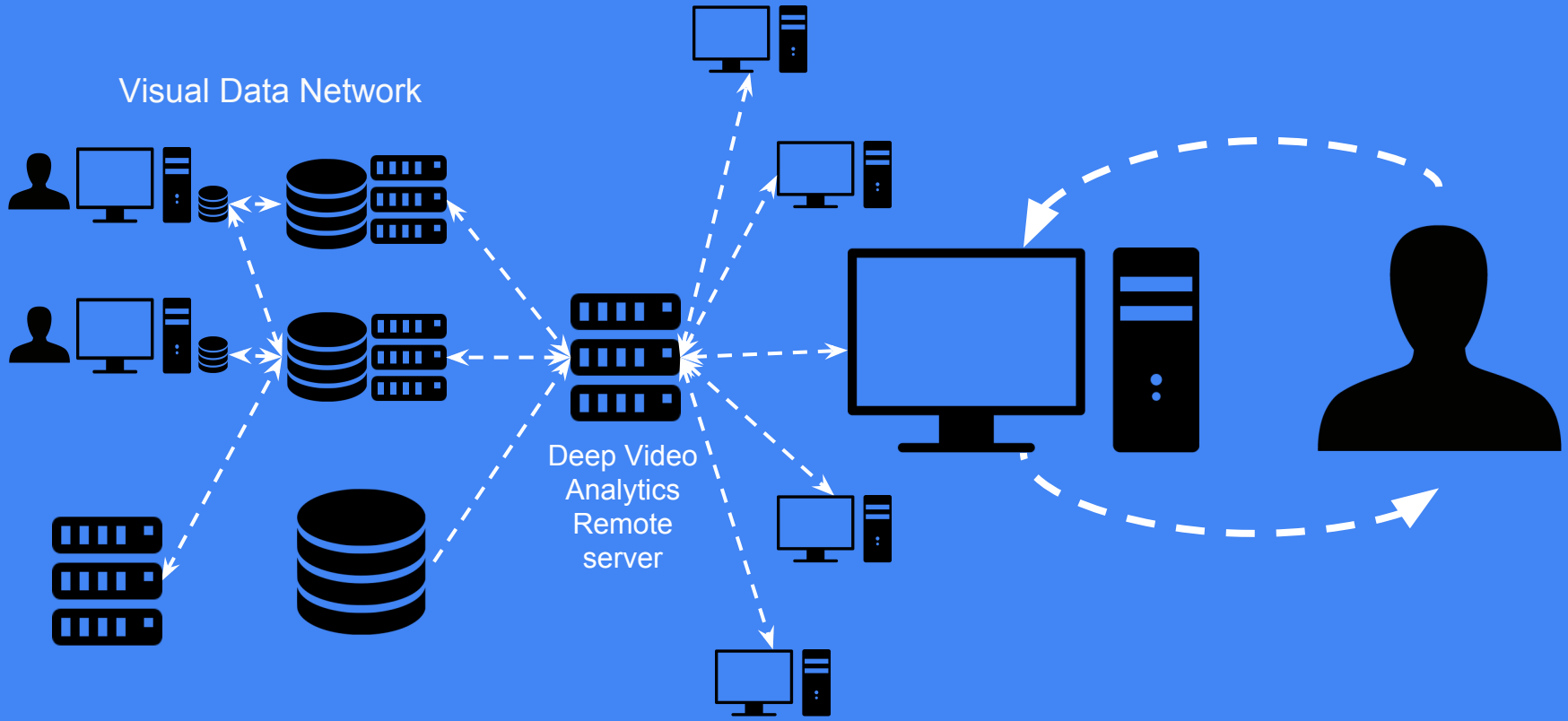
Sharing data using Visual Data Network

Import & export new datasets / annotations
share with other users

Visual Data Network



Flexible deployment: local & remote server



Deep Video Analytics
Design goals

- Usable by non-researchers
- Visual Search as a “Primary User Interface”
- Users can provide data easily (via upload, youtube-dl, annotation UI etc.)
- Batteries-included approach with an indexing and detection pipeline
  - Tensor Flow Inception v3, Single Shot Detector trained on VOC & YOLO 9000
  - Face detection / alignment / recognition
  - More algorithms such Text detection, Audio features.
- Pre-indexed datasets from different domains can be quickly loaded
- Can be easily customized by developers & researchers.

Deep Video Analytics
Technical goals

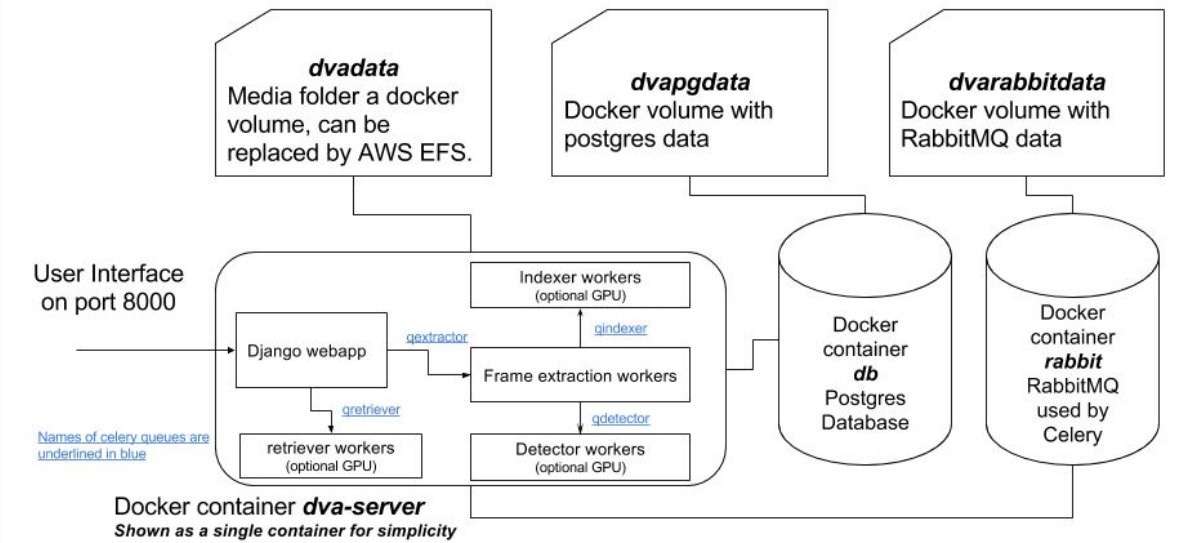
- Useful without having to write code or config
- Works on machines with and without GPUs
  - Works (albeit slowly) without a GPU, tested on Linode VPS with 8Gb RAM & 4 Cores
- Handles uploads and continuous index updates
- Data can be easily imported, exported and shared
- Can be easily modified by technical users
  - E.g. Adding more operations to processing pipeline
- Can be scaled out by adding more GPUs / Machines

Deep Video Analytics
Frameworks & technologies used

- Django, Postgres, Celery, RabbitMQ, Tensorflow, Docker, all are widely used.

Emulating datacenter on a machine
Docker, Docker-compose, Nvidia-docker

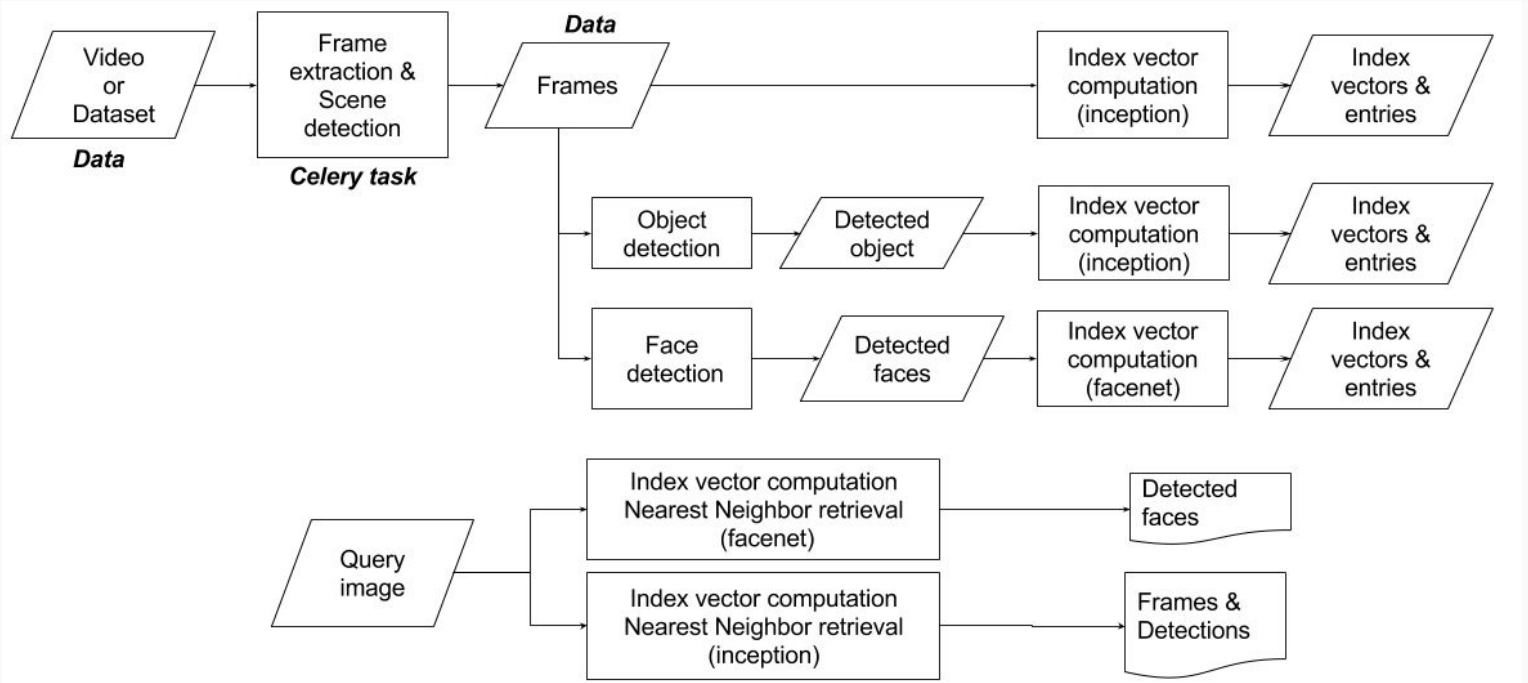
Docker enables same codebase across all configurations {a laptop, multi-GPU machine, datacenter} .



Data model

- Video / Dataset
  - Video or collection of images
- Frame
  - Single frames or image
  - Must have parent Video / Dataset
- Detection
  - A bounding box in a frame/image
  - Algorithm, confidence & label
- Annotation
  - Name, Metadata
  - Bounding box in frame / detection
- Query
  - Optional user, time
- QueryResult
  - Parent query
- Task Event
  - Outcome of processing on a particular video/dataset or a query
- IndexEntry
  - Indexing algorithm (inception, facenet, etc.)
  - Indexed object (frames, specific object)
  - Entry and numpy features filename

Flowchart
Video & Query processing



Code organization:
dvaapp & dvalib

dvaapp: a django app/project

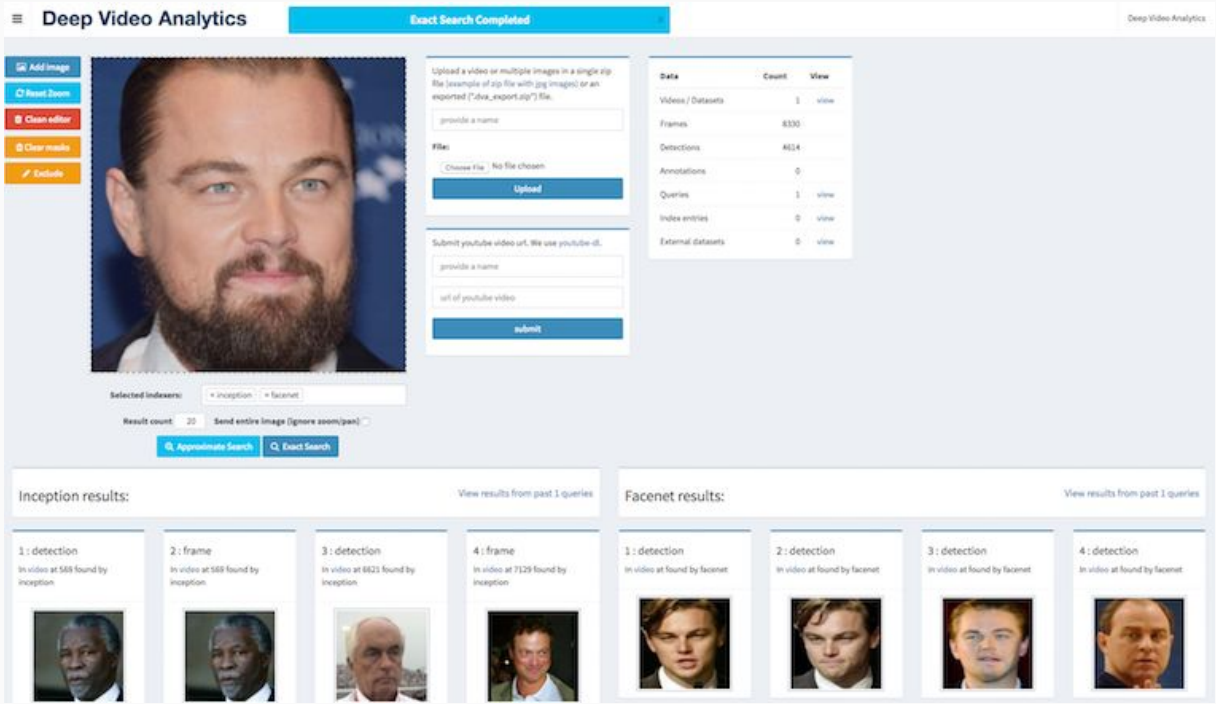
- Handles UI and data processing
- Data model
  - Video, Frame, Detection
  - Query, QueryResult
  - Event, etc.
- A set of celery tasks
  - Extract frames / process video
  - Perform indexing
  - Perform detection
- Uses dvalib to carry out tasks

dvalib: library for handling algorithms

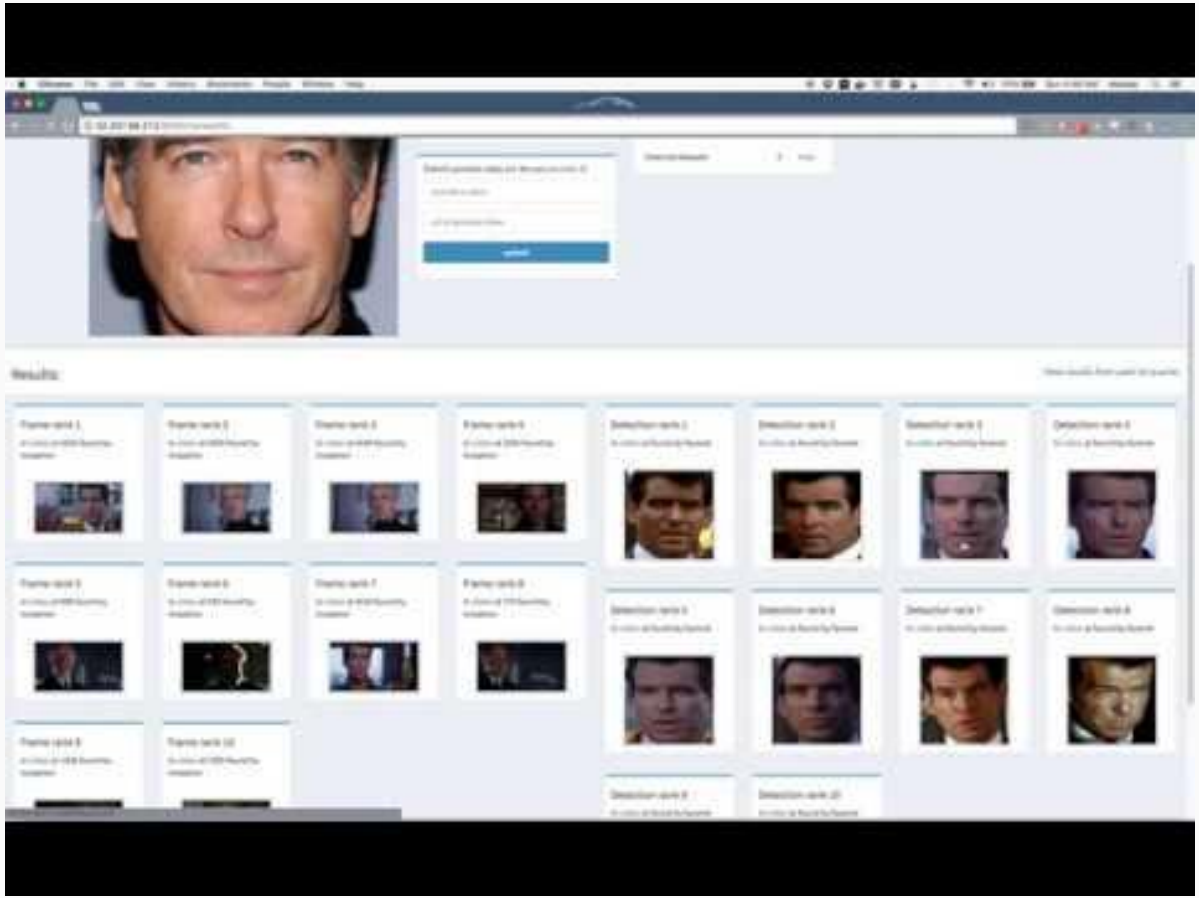
- A database & celery agnostic library
- Interface with Tensor Flow & Pytorch for
  - extraction
  - detection
  - indexing
- Usable without having a running django instance, but designed to interface with it. E.g. assumptions regarding layout of directories containing videos, frames etc.

User Interface:

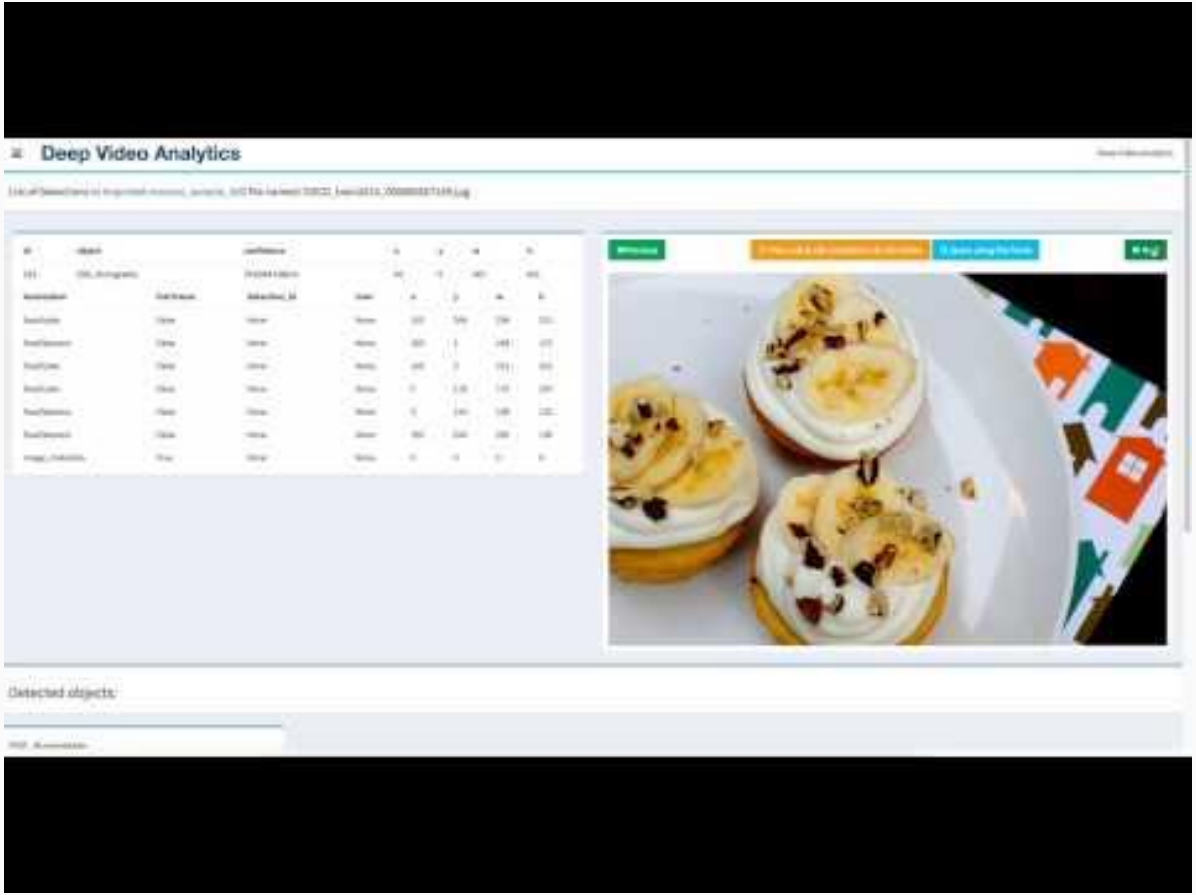
Search across frames + detections (faces, etc.)



Demo Version Alpha 1, 15th March 2016



Demo Version Alpha 2, 7th April 2017



Open questions:
A work in progress

- How to rank results using auxiliary information?
- How to balance fast/static vs slow/dynamic indexes?
- How to incorporate text data extracted from images?
- Learning from annotations?
- Real time plug-in that bypasses queue based system?
- An Android / iOS frontend app for data acquisition?

Thanks!

Contact me:

akshayubhat@gmail.com
www.akshaybhat.com