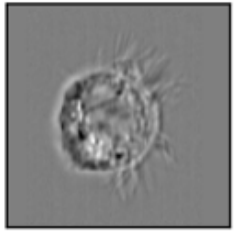
Label-free classification of ciliated cells using Deep Learning

Ketil Tvermosegaard

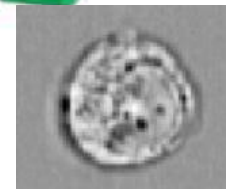
PSI Webinar

Acknowledgements:
Steven Barrett, Gareth Wayne, James Porter, Luke Markham, Sam Bates, Paul Cooper

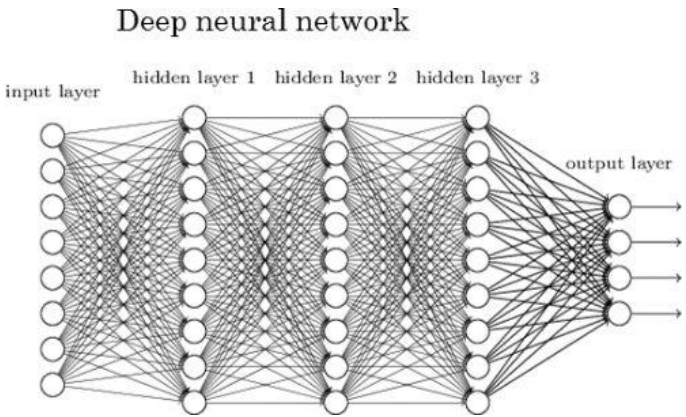
One-slide summary



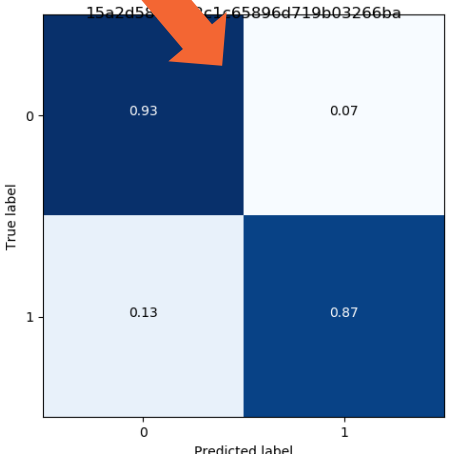
ciliated
(hairy)



non-ciliated
(not hairy)

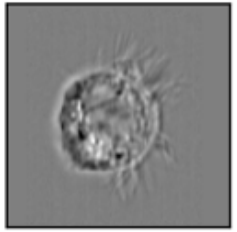


Tessella
•ALTRAN GROUP

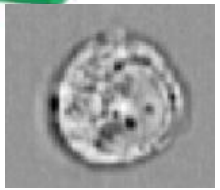


One-slide summary

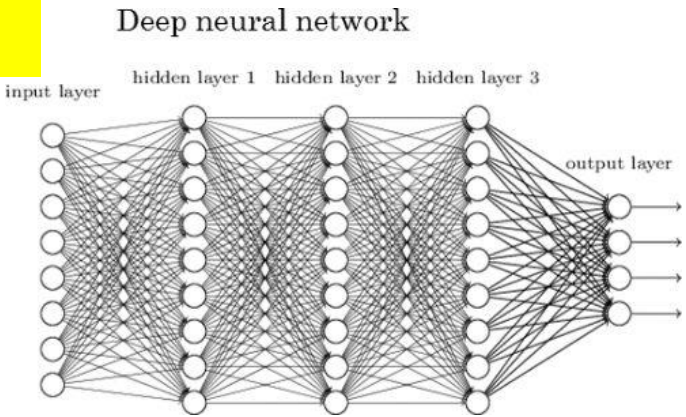
Gareth Wayne (Novel Human Genetics)
New “image cytometer” (ca. £500K)
Produces pictures of cells (thousands).
Can we use ML to label cells?



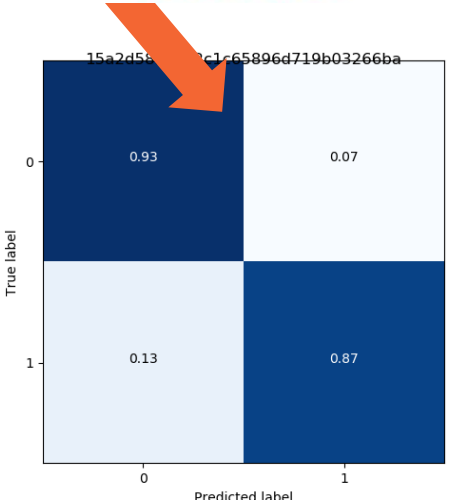
ciliated
(hairy)



non-ciliated
(not hairy)

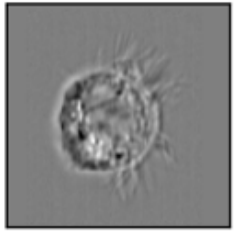


Tessella
•ALTRAN GROUP

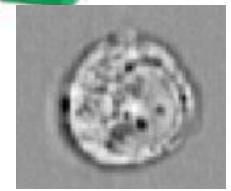


One-slide summary

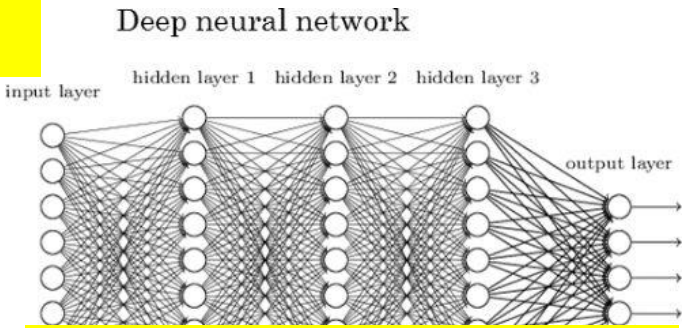
Gareth Wayne (Novel Human Genetics)
New “image cytometer” (ca. £500K)
Produces pictures of cells (thousands).
Can we use ML to label cells?



ciliated
(hairy)

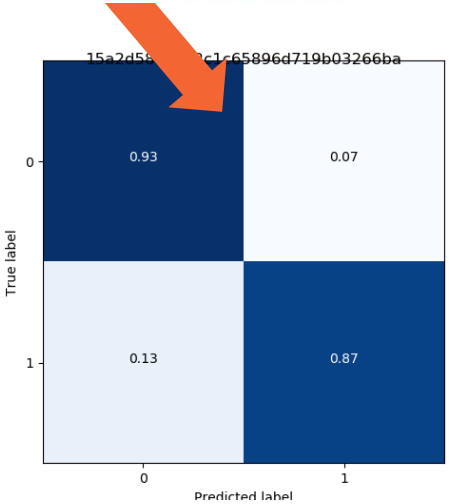


non-ciliated
(not hairy)



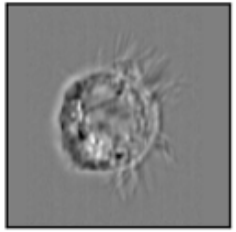
Ketil Tvermosegaard and Steven Barrett (Research Statistics)
Sounds like a Deep Learning problem.
But can we access images?
And will DL work?

Tessella
•ALTRAN GROUP

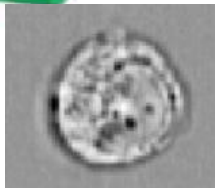


One-slide summary

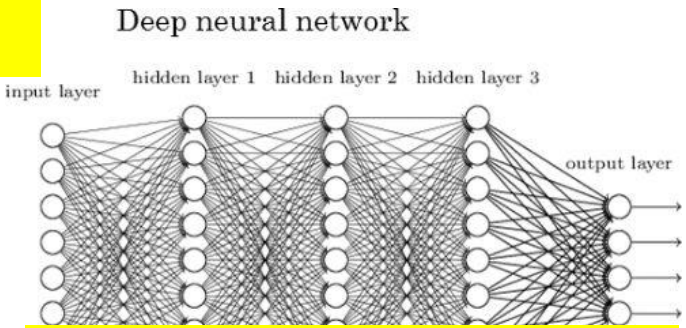
Gareth Wayne (Novel Human Genetics)
New “image cytometer” (ca. £500K)
Produces pictures of cells (thousands).
Can we use ML to label cells?



ciliated
(hairy)

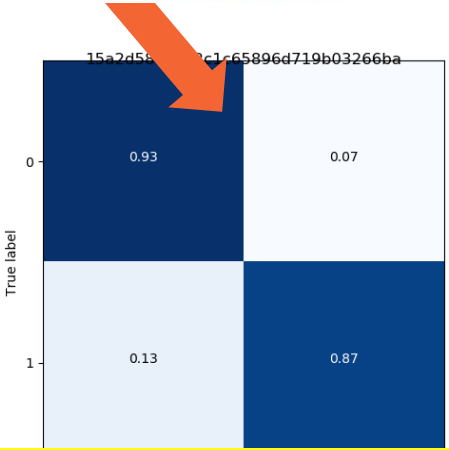


non-ciliated
(not hairy)



Ketil Tvermosegaard and Steven Barrett (Research Statistics)
Sounds like a Deep Learning problem.
But can we access images?
And will DL work?

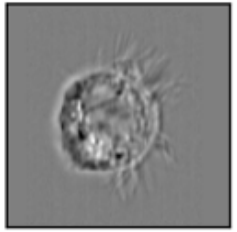
Tessella
•ALTRAN GROUP



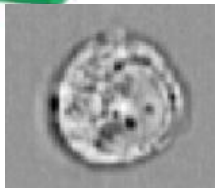
Paul Cooper, Sam Bates, Luke Markham (Tessella)
Improve Ketil and Steven’s POC
Build prototype code for production

One-slide summary

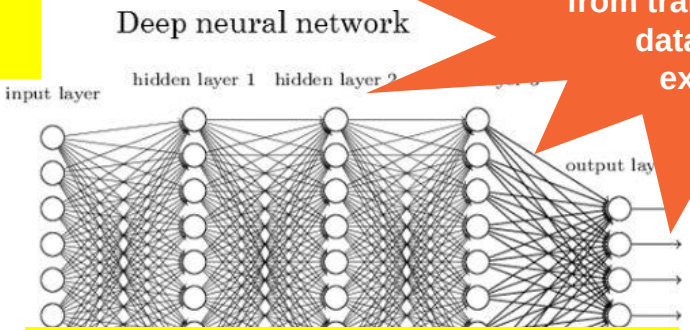
Gareth Wayne (Novel Human Genetics)
New “image cytometer” (ca. £500K)
Produces pictures of cells (thousands).
Can we use ML to label cells?



ciliated
(hairy)

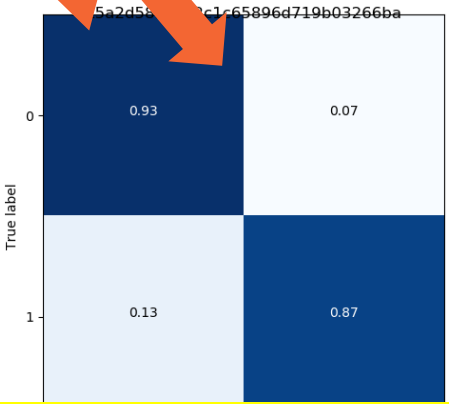


non-ciliated
(not hairy)



Ketil Tvermosegaard and Steven Barrett (Research Statistics)
Sounds like a Deep Learning problem.
But can we access images?
And will DL work?

Actual confusion matrix from trained network on data from NEW experiment!



Paul Cooper, Sam Bates, Luke Markham (Tessella)
Improve Ketil and Steven’s POC
Build prototype code for production

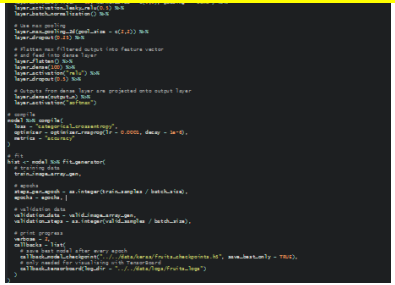
Key Learnings

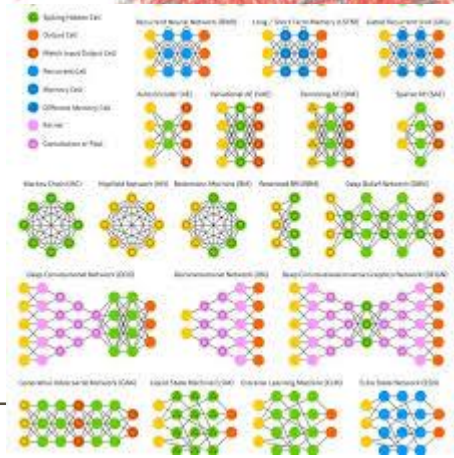
Key Learnings

DL is not hard to use

POC network took about 100 lines of R-code using keras package

And this code was mostly lifted from a tutorial example used to recognise images of fruit



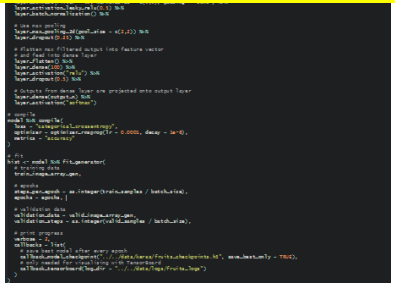


Key Learnings

DL is not hard to use

POC network took about 100 lines of R-code using keras package

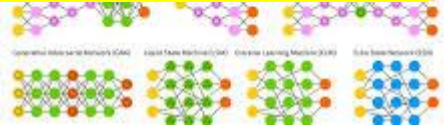
And this code was mostly lifted from a tutorial example used to recognise images of fruit



DL is hard to use well

Essentially an infinite-dimensional optimisation problem
- architecture
- hyper-parameters
- pre-processing

Solved by Tessella's "test bench" approach

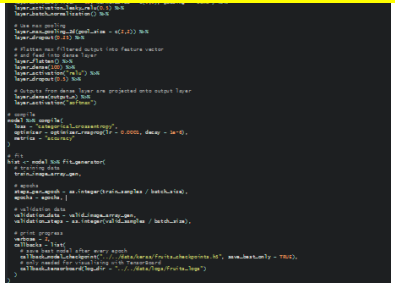


Key Learnings

DL is not hard to use

POC network took about 100 lines of R-code using keras package

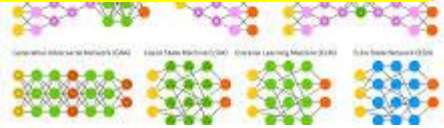
And this code was mostly lifted from a tutorial example used to recognise images of fruit



DL is hard to use well

Essentially an infinite-dimensional optimisation problem
- architecture
- hyper-parameters
- pre-processing

Solved by Tessella's "test bench" approach



Is DL right for you?

Is the data the **right kind**?
- After a lot of work, we got access to images

Do you have **enough**?
- 10s of thousands of **manually** labelled images

Is your problem actually a DL problem?
- We had a **clear visual phenotype** (recognisable to non-expert)

Timeline of pre-TAP work

Problem Definition
- Derived features
- Proprietary file format

ML on features?
- Failed

“Crack” file format?
- Success

Contact Tessella
- Started Tessella Analytics Partnership (TAP) project

POC
- DL sufficient to proceed

Deep Learning?
- Fruits

.....

Timeline of pre-TAP work

"My computer science definition of **progress**: generating new error messages"

Problem Definition
- Derived features
- Proprietary file format

ML on features?
- Failed

"Crack" file format?
- Success

.....

Deep Learning?
- Fruits

POC
- DL sufficient to proceed

Contact Tessella
- Started Tessella Analytics Partnership (TAP) project

Timeline of TAP work

Timeline of TAP work

Build

Refine

Apply

- Start naïve (fruits)
- Network “test bench”
- Test architecture and hyper-params

Timeline of TAP work

Build

Refine

- Data augmentation
- Architecture
- Optimisation

Apply

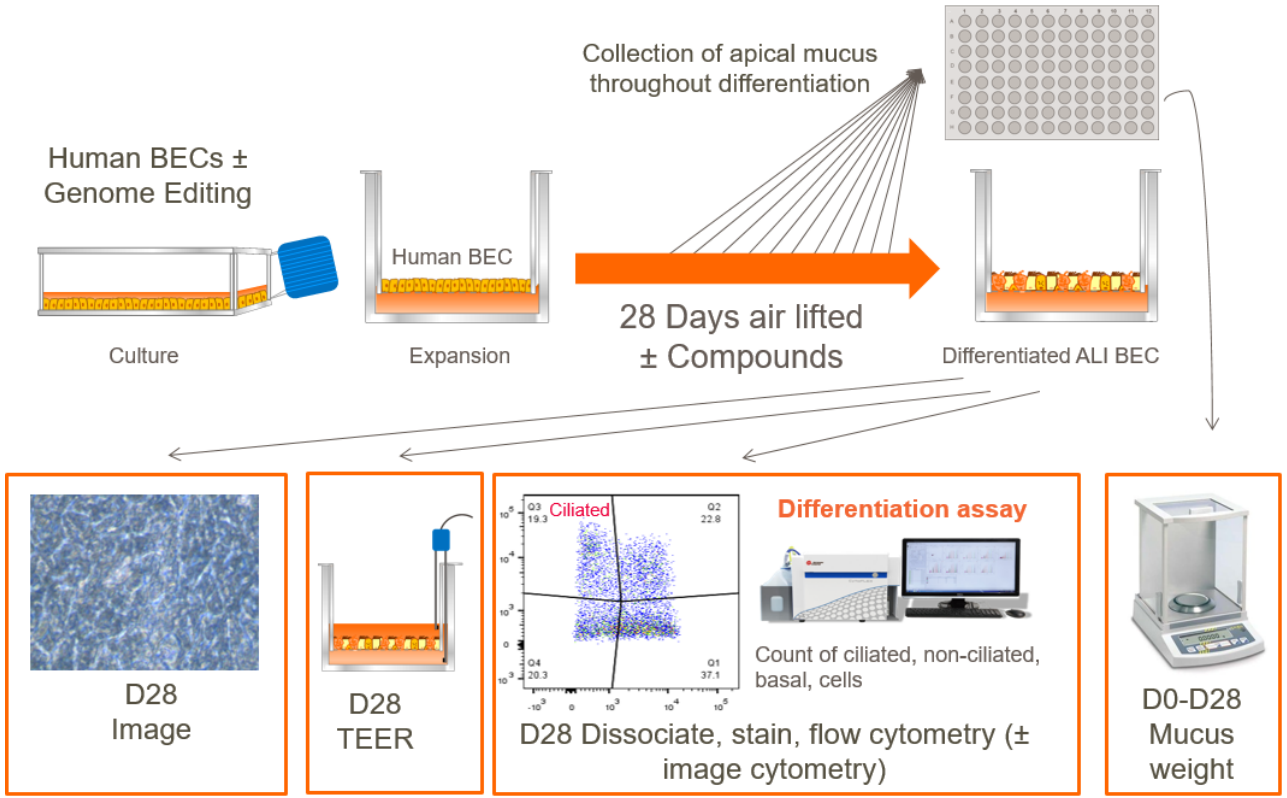
Timeline of TAP work

Build → Refine → Apply

- New data
- New cell types
- Mis-labels

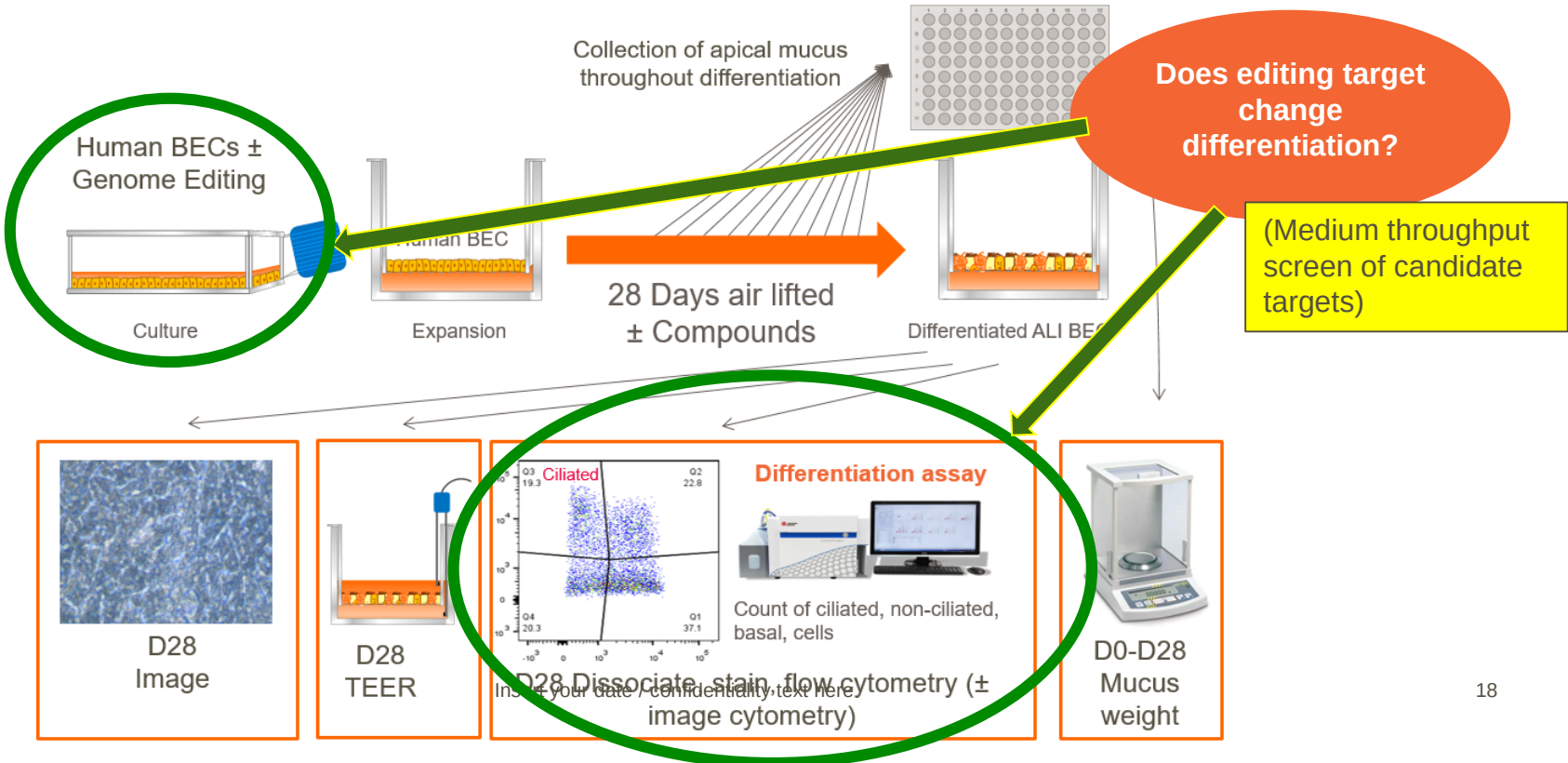
Epithelial Differentiation screening

Source: Gareth Wayne

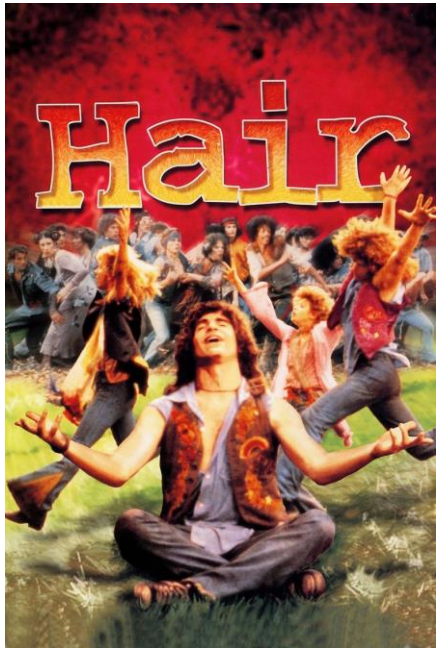


Epithelial Differentiation screening

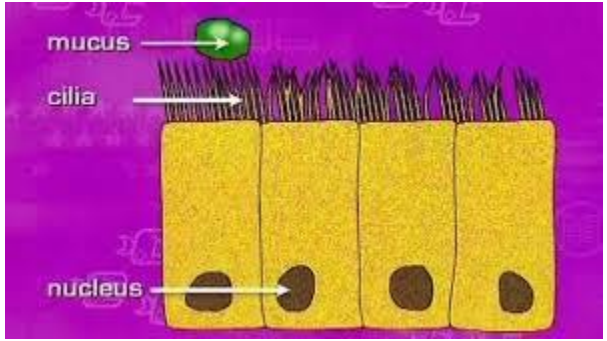
Source: Gareth Wayne



Ciliated cells

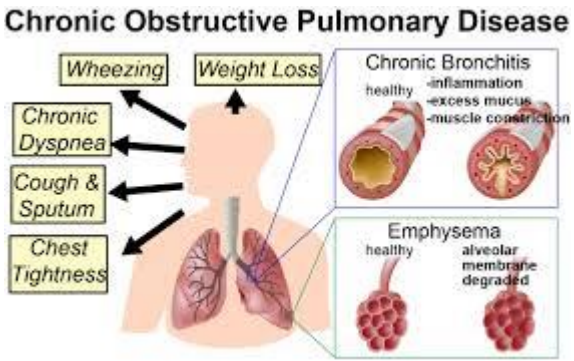


… on cells. What is it good for?



transportation
protection
secretion

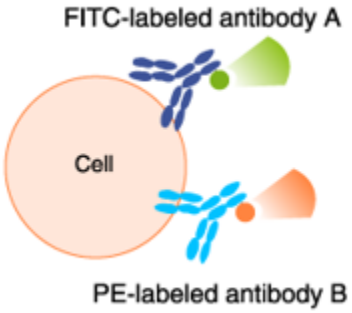
Important in respiratory indications like…

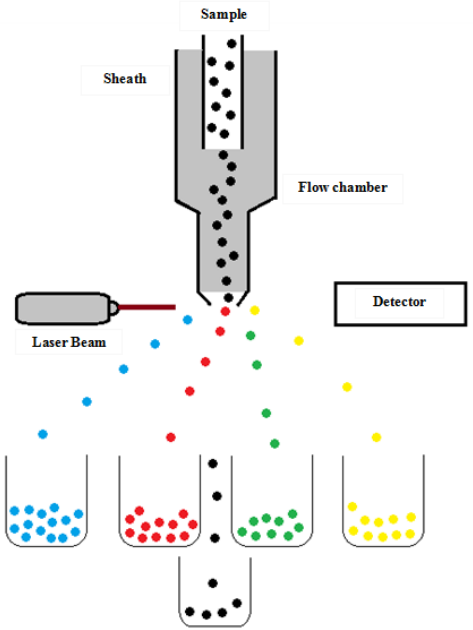


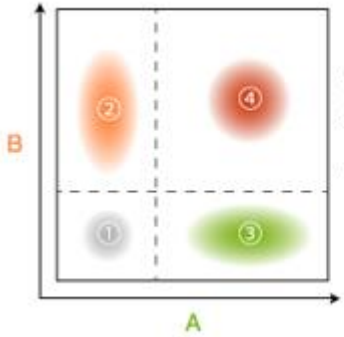
biophysics.org

Flow cytometry (FACS)

In principle, flow cytometry is easy…

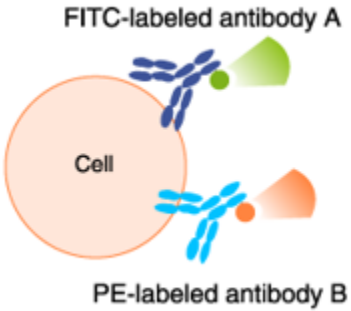




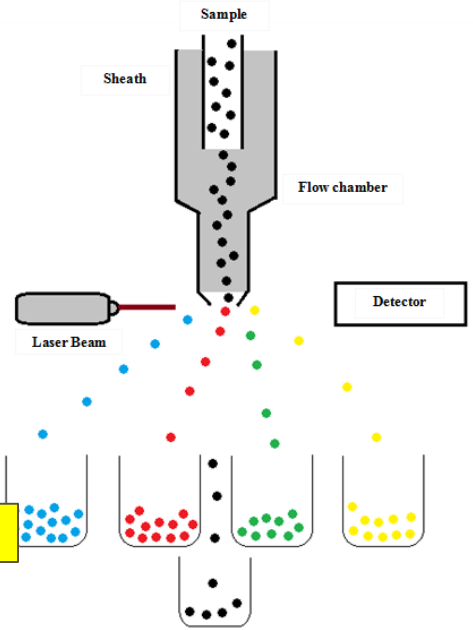


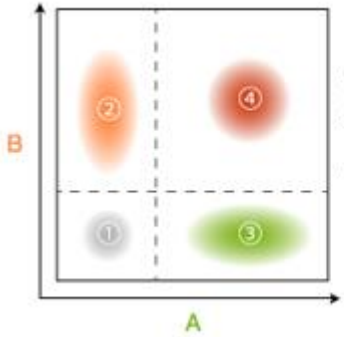
Flow cytometry (FACS)

In principle, flow cytometry is easy…



Label cells with fluorophore-antibody pairs

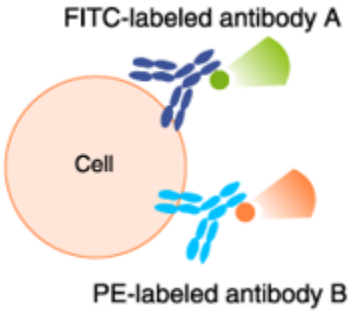


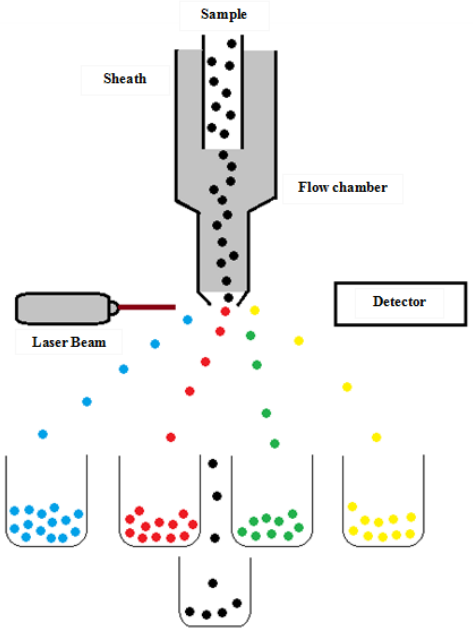


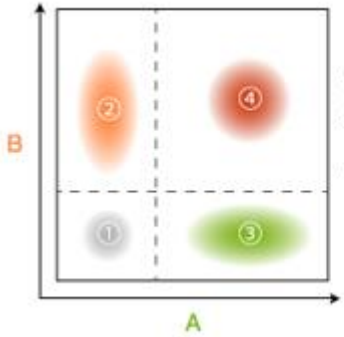
1) Cells not expressing either A or B
2) Cells expressing only B
3) Cells expressing only A
4) Cells expressing both A and B

Flow cytometry (FACS)

In principle, flow cytometry is easy…



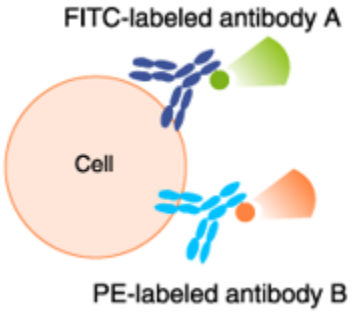


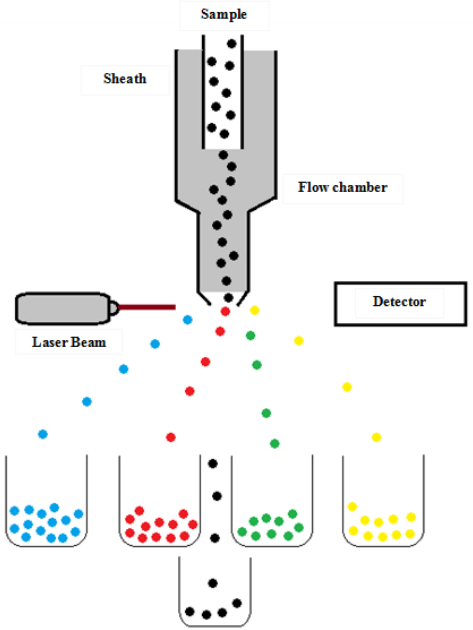


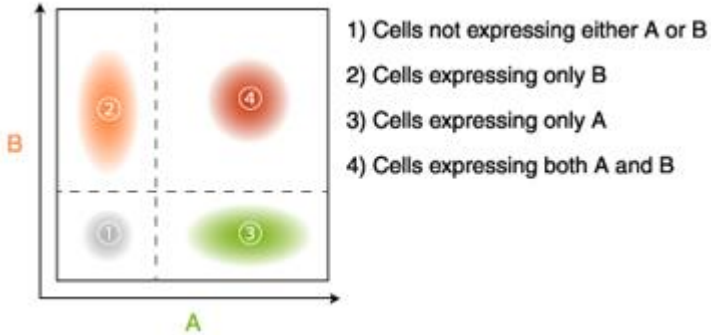
Use laser to read wavelength of light emitted by each cell

Flow cytometry (FACS)

In principle, flow cytometry is easy…



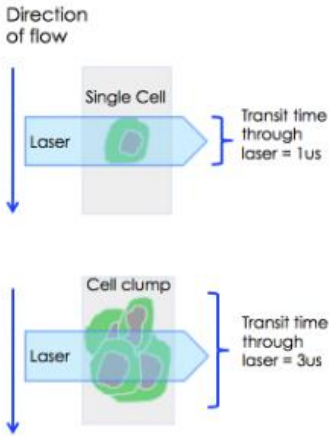




Classify cells based on label expression

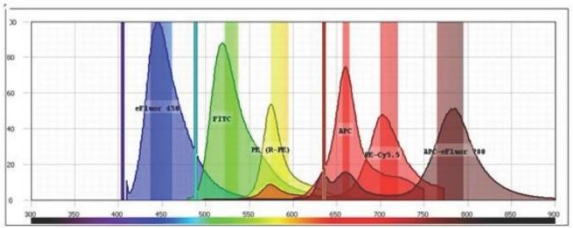
Flow cytometry (FACS)

But in practice…



Do we really get a single cell at a time?

- Compensation corrects for spillover between fluorochrome emission spectra.



- Compensation is essential for multicolor panels

Fluorophores overlap!

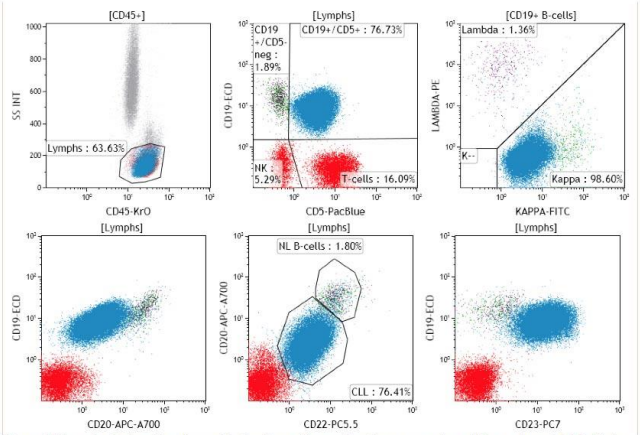


Figure 1: Normal polyclonal B-cells provide the internal "normal" antigen expression which results on CD19+(dim), CD20+(dim), CD22+(dim), CD23+(bright) and dim expression for kappa.

What is the appropriate “gating” to identify cell types?
+
Gating is sequential!

Image Flow Cytometry

… to the rescue?

– Image Flow Cytometry = Flow cytometry + Cell imaging camera

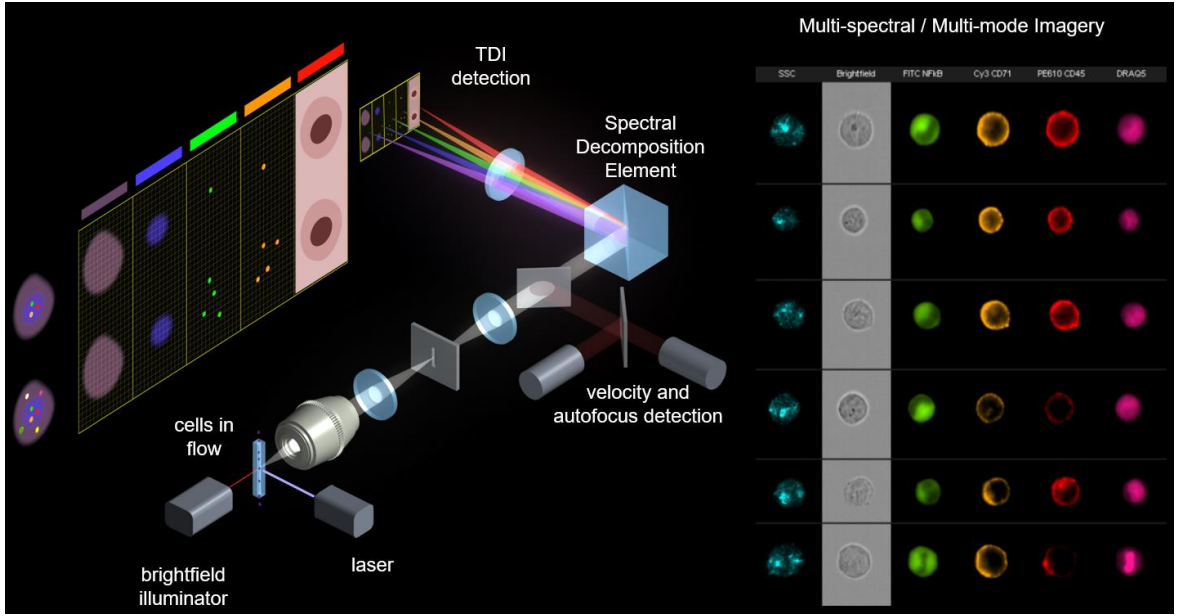
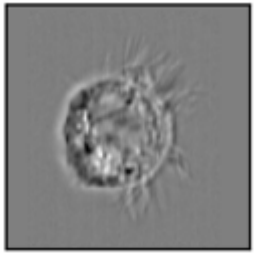


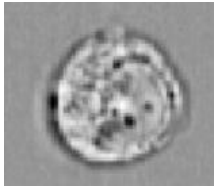
Image from Merck Millipore / Amnis

The “scientific problem”

- Scientist using FACS to determine if epithelial cells were ciliated (“hairy”) or not
- Using single cell images (Image Flow Cytometry) to validate findings
- Validation not always consistent with FACS

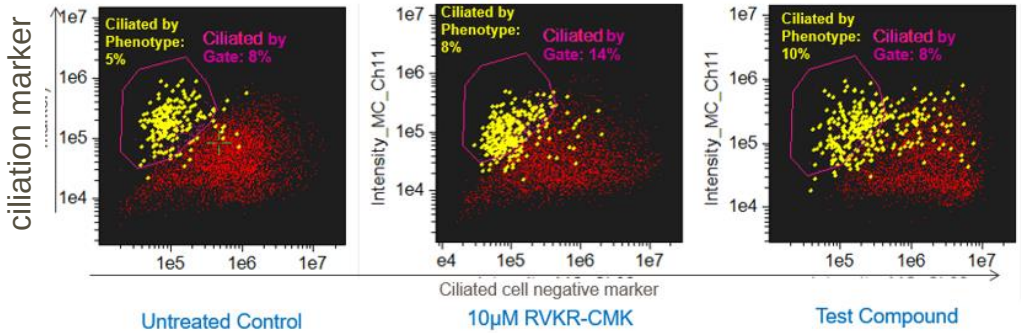


ciliated



non-ciliated

representative images



The “statistical problem”

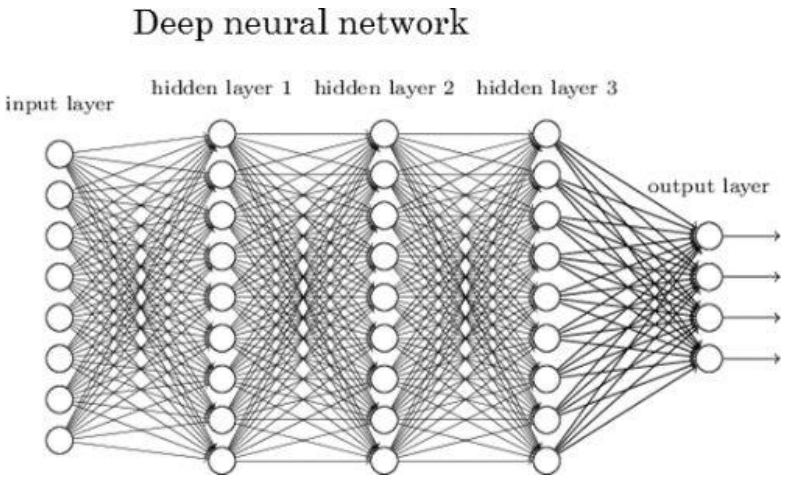
- Many thousands of images (5,000 – 10,000 cells in a well, approx. 30 wells to a plate)
- Derived numerical features available (sphericity, diameter, etc)
- Image files in proprietary file format
- Obvious visual phenotype

Deep Learning

– Special case of (Artificial) Neural Network, characterized by having multiple layers

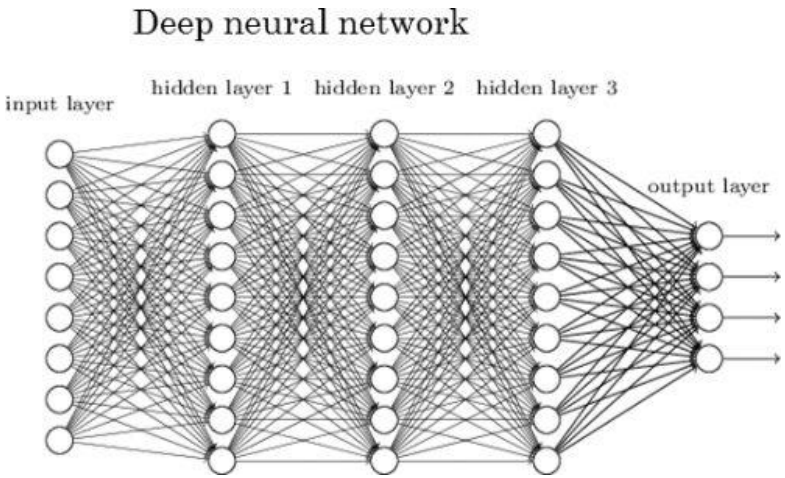
Deep Learning

– Special case of (Artificial) Neural Network, characterized by having multiple layers



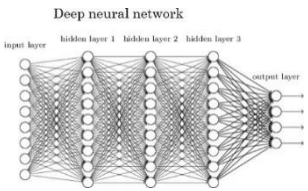
Deep Learning

- Special case of (Artificial) Neural Network, characterized by having multiple layers
- Many kinds of layers. We use **activation, convolutional, pooling, dropout**



Deep Learning

- Special case of (Artificial) Neural Network, characterized by having multiple layers
- Many kinds of layers. We use **activation, convolutional, pooling, dropout**



<u>Activation</u>: each node has (scalar-valued) output

$$g(w'x + b) = g(\Sigma_{i=1}^{n} w_i x_i + b) \text{ , with e.g. } g = \tanh$$

 # Parameters: (one weight vector *w* of same length as *x* plus one single *bias* scalar) X (# nodes)
Intuition: combination of linear transformation and (softly) step-like functions = **flexible function approximation**
NB: "The Universal Approximation Theorem"

I.e., each node =

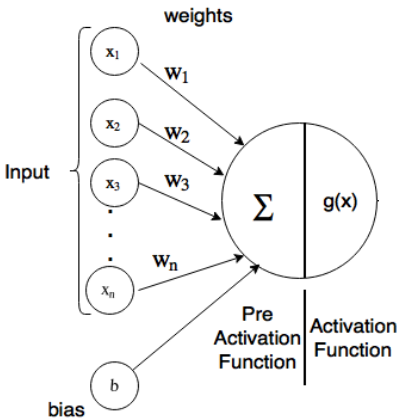
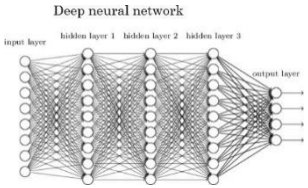


image: https://medium.com/@akankshamalhotra24/tutorial-on-feedforward-neural-network-part-1-659eeff574c3

Deep Learning

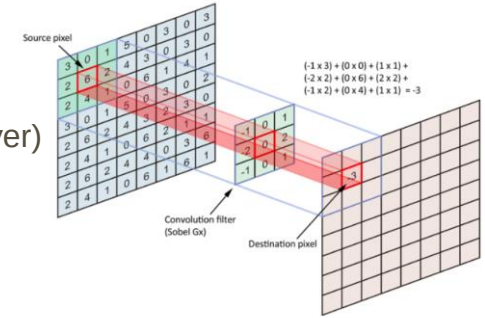
- Special case of (Artificial) Neural Network, characterized by having multiple layers
- Many kinds of layers. We use **activation, convolutional, pooling, dropout**



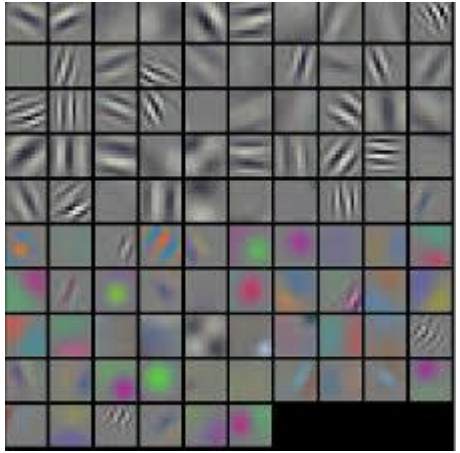
Convolution: each node has (array-valued) output.
A "filter" array is multiplied element-wise onto the input array and the sum is taken.
This filter is run across the entire input array yielding a new (smaller) array.
Example output for 2-dimensional input:

$$Z_{i,j} = \Sigma_{a=1}^{k} \Sigma_{b=1}^{k} F_{a,b} x_{i+a,j+b}$$

 # Parameters: (#cells in filter) X (#nodes in layer)
Intuition: each node can learn a "feature".
E.g. circles, horizontal lines, etc.

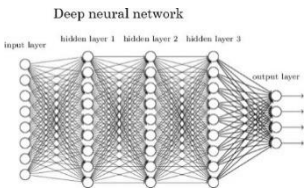


The filter slides over the input and performs its output on the new layer. — Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2



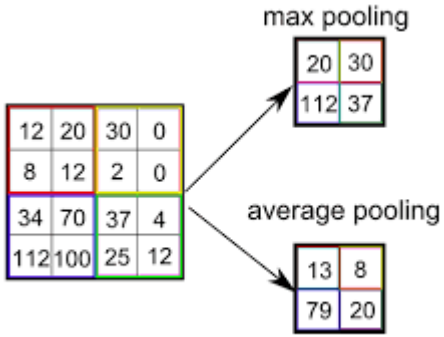
Deep Learning

- Special case of (Artificial) Neural Network, characterized by having multiple layers
- Many kinds of layers. We use **activation, convolutional, pooling, dropout**



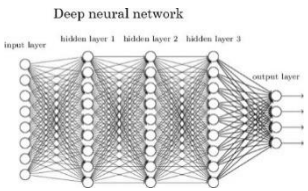
Pooling: each node has (array-valued) output.
The input array is divided into a grid and a simple “pooling function” is applied to all the cells in each “grid chunk”.

Parameters: none
Hyper-parameters: size of filter, step size
Intuition: data compression + trying to extract “salient features” (data might be “grainy”)



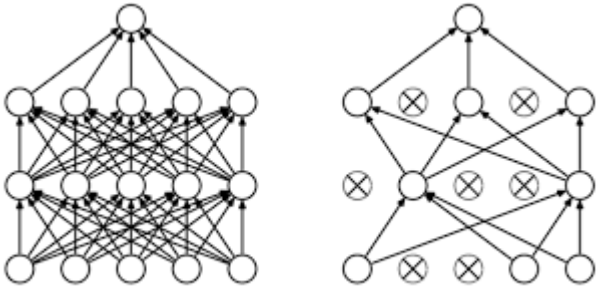
Deep Learning

- Special case of (Artificial) Neural Network, characterized by having multiple layers
- Many kinds of layers. We use **activation, convolutional, pooling, dropout**



<u>Dropout</u>: Every iteration of training, randomly drop a fixed proportion of nodes in the layer

 # Parameters: none
Hyper-parameters: dropout rate / number
Intuition: ”Robustification” against dominating/correlated features.
Similar in spirit to randomly dropped features in random forests.



(a) Standard Neural Net (b) After applying dropout.

Deep Learning

– Special case of (Artificial) Neural Network, characterized by having multiple layers

Gets a special name because **It Works!**

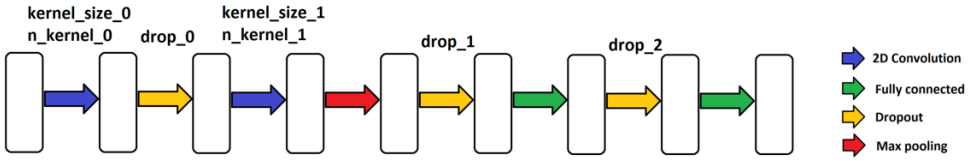
ImageNet Challenge 2012
- Vast improvement on earlier technologies
- Many examples followed
  (Google translation, speech recognition, etc.)

Tessella Analytics Partnership Project

Process & Progress

Architecture & Hyperparameters

- Systematic experimentation to optimise deep-learning architecture and hyper-parameters
- Ensembling of several networks



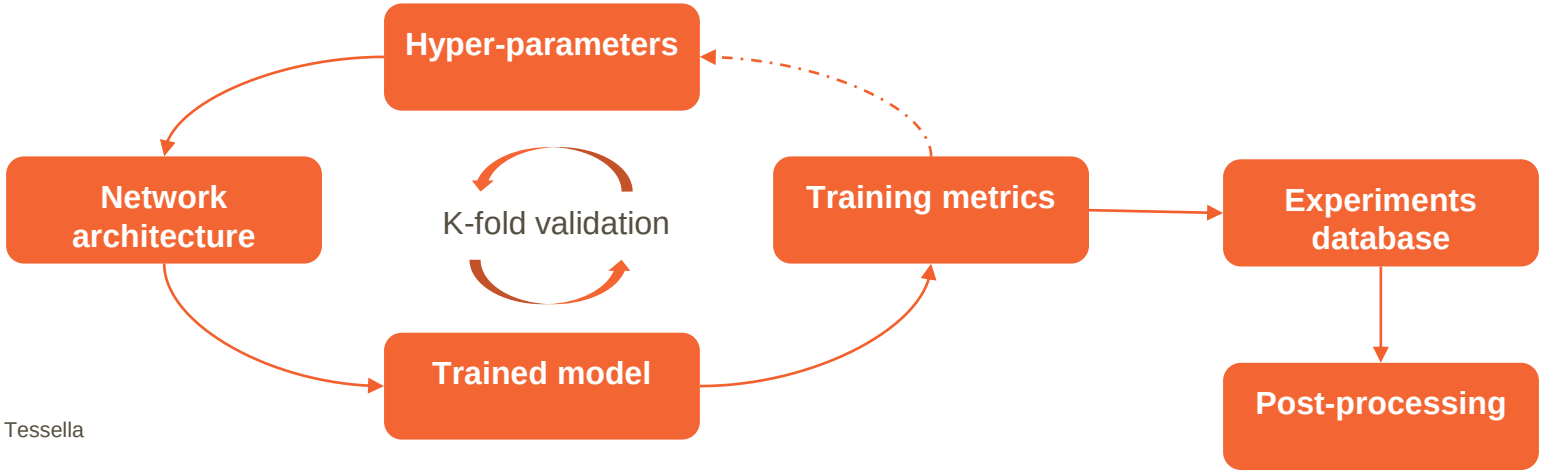
Training-set Improvement

- Tessella deep-learning image analysis expertise critical to:
  - Image pre-processing (standardisation of raw cell images)
  - Image augmentation (perturbation of input to increase volume/diversity of training set)
- ‘Ground truth’ improvement (re-presentation of false+/-’s to human experts)
- Further human image labelling (more experiments, other primary cell donors)

source: Steven Barrett

Test bench, conceptually

- Facilitates experimentation with different network architectures
- Iterate over hyper-parameters
- Record results in a database-style format



source: Tessella

Test bench, concretely

Simple, but highly effective

– “Wrapper” function which takes hyperparameters + architecture as input and returns uniquely ID’ed output (input + performance metrics)

```python
from src.nn_models.meta_model_container_class import MetaModelContainer
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPooling2D, Flatten, Dropout
from keras.optimizers import rmsprop
from src.constants import ARCH_CONST, PREPROC_CONST


class Model_24(MetaModelContainer):
    """

    """
    def __init__(self):
        self.params = {"n_epoch": 300,
                       "n_batch": [64, 128, 256],
                       "learn_rate": [2E-4, 5E-4, 1E-3],
                       "n_kernel_0": 16,
                       "kernel_size_0": 3,
                       "n_kernel_1": 12,
                       "kernel_size_1": 3,
                       "drop_4": [0, 0.1],
                       "drop_0": [0, 0.1],
                       "drop_1": 0.25,
                       "drop_3": 0.25,
                       "drop_2": 0.25,
                       "n_dense_1": 48,
                       "rotation_range": 90,
                       "shift_range": 0.05,
                       "shear_range": 10,
                       "zoom_range": 0.1,
                       "horizontal_flip": True,
                       "vertical_flip": True,
                       "brightness_range": None,
                       "fill_mode": "constant",
                       "cval": PREPROC_CONST["background"]}
```

source: Tessella

Understanding the effects of hyper-parameters

– A snapshot of an architecture’s training history:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	hash	drop_0	drop_1	drop_2	kernel_size	kernel_size	learn_rate	n_batch	n_dense_1	n_epoch	n_kernel_0	n_kernel_1	validation_cm	test_acc
2	c1c2516e737	0	0.5	0.5	3	3	0.001	32	48	50	8	8	[[0.82_ 0.18]_ [0.23_ 0.77]]	0.828369
3	fdaf70bd438	0	0.5	0.5	3	3	0.001	32	64	50	8	10	[[0.87_ 0.13]_ [0.26_ 0.74]]	0.822695
4	1532dc2c918	0	0.5	0.5	3	3	0.001	32	32	50	16	12	[[0.89_ 0.11]_ [0.33_ 0.67]]	0.822695
5	92a859840d	0	0.5	0.5	3	3	0.001	32	64	50	16	12	[[0.8_ 0.2]_ [0.19_ 0.81]]	0.821277
6	c9eb8da2197	0	0.5	0.5	3	3	0.001	32	48	50	16	12	[[0.75_ 0.25]_ [0.15_ 0.85]]	0.819858
7	43fc40831ae	0	0.5	0.5	3	3	0.002	32	64	50	8	10	[[0.75_ 0.25]_ [0.21_ 0.79]]	0.81844
8	54c32ca5aa3	0	0.5	0.5	3	3	0.001	32	48	50	8	10	[[0.89_ 0.11]_ [0.3_ 0.7]]	0.812766
9	ba931dcf2bd	0	0.5	0.5	3	3	0.002	32	32	50	16	8	[[0.79_ 0.21]_ [0.16_ 0.84]]	0.812766
10	039b9448937	0	0.5	0.5	3	3	0.002	32	32	50	16	12	[[0.83_ 0.17]_ [0.24_ 0.76]]	0.809929
11	f6b16f295f71	0	0.5	0.5	3	3	0.0005	32	64	50	16	12	[[0.89_ 0.11]_ [0.35_ 0.65]]	0.808511
12	1959138c081	0	0.5	0.5	3	3	0.001	32	48	50	16	8	[[0.66_ 0.34]_ [0.2_ 0.8]]	0.805674
13	[illegible]	0	0.5	0.5	3	3	0.001	32	48	50	8	12	[[0.86_ 0.14]_ [0.3_ 0.7]]	0.805674

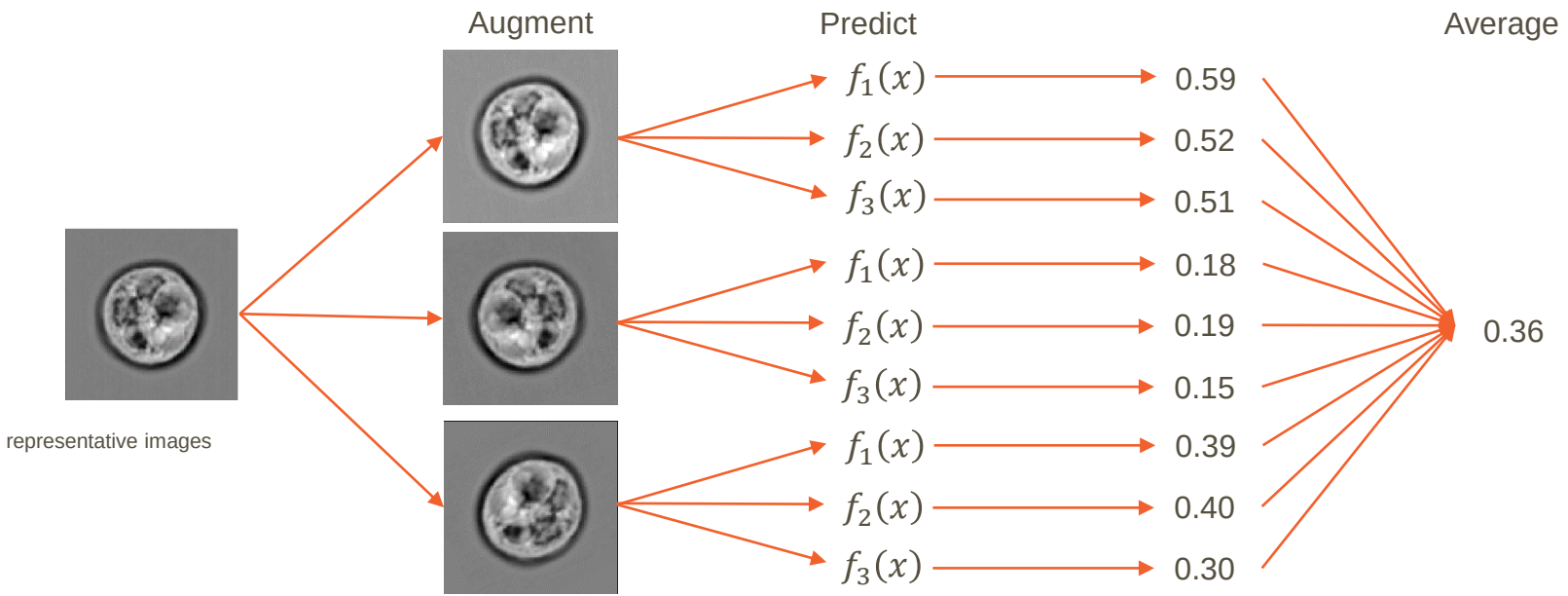
– Dropout between first and second convolutional layers highly reduces performance

– The number of filters in the fully connected and convolutional layers is irrelevant

Neural Network Ensembles

source: Tessella

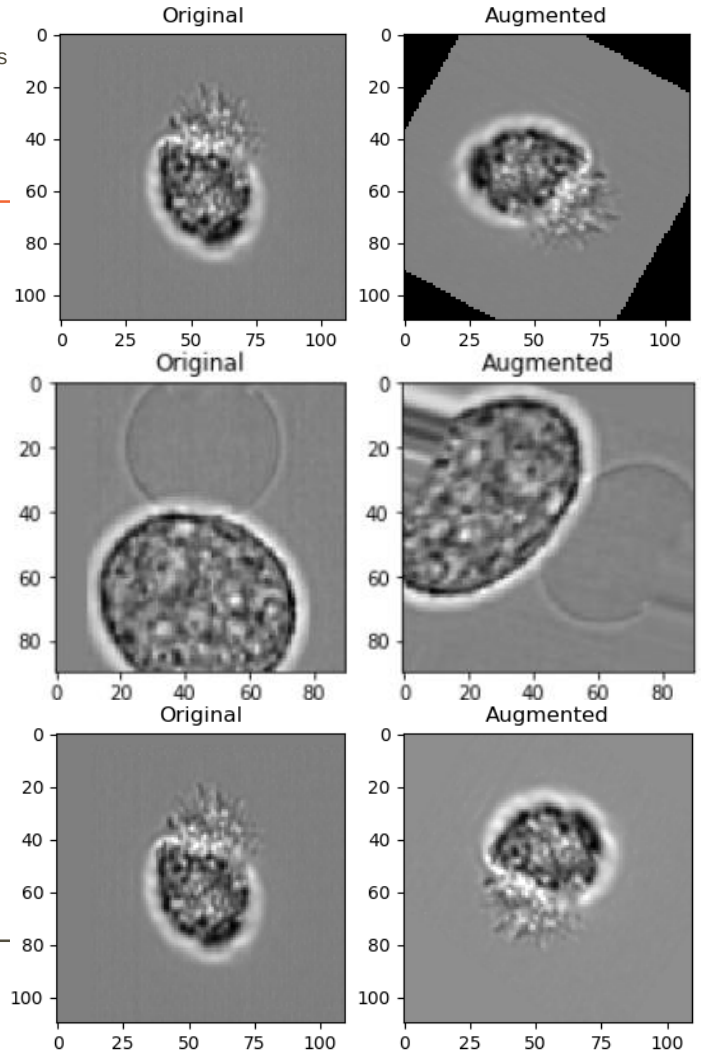
- Combine the results of multiple trained classifiers in a weighted voting system
- Each classifier has learnt different features and found slightly different optima
  - Average of predictions more robust to unimportant differences between optima



Augmentation exposed areas:

source: Tessella

representative images



Augmentation exposes areas which were not imaged

How to fill in the gaps?

- Initially tried nearest pixel method
- Deforms cells in some cases

Solution

- Estimate background
- Standardise image by piecewise linear transform, setting background to fixed value
- Fill exposed pixels with the same value

Hit Validation in Modulation of Ciliation

Ketil Tvermosegaard, Research Statistics

Situation

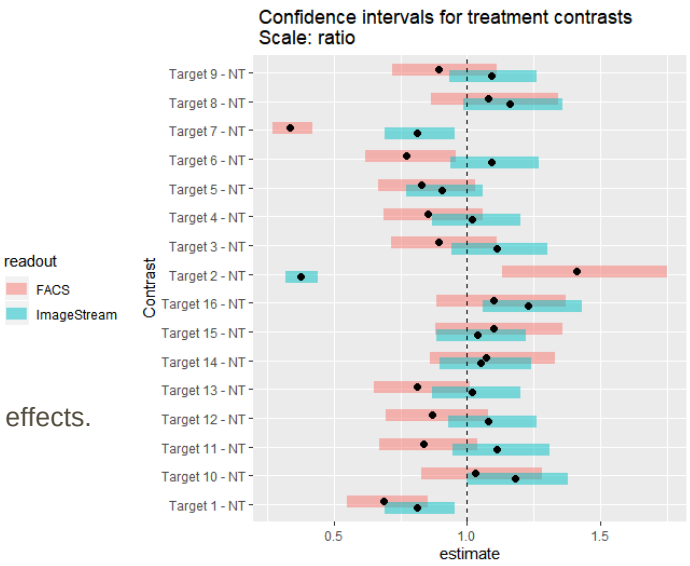
Gene KO experiment with two main readouts

- marker-based flow cytometry (**FACS**)
- image flow cytometry (**ImageStream**), processed with Deep Learning

Purposes
(i) to investigate whether KO of the hits modulates ciliation
(ii) whether FACS and ImageStream concur

Action
Experimental design was provided: A **complete block design** with three blocks (each block a separate replication of the full KO experiment).
Linear mixed effects models were fitted separately for each endpoint, to control for block effects.

Impact
Some **strong disagreements** between FACS and ImageStream.
This was expected, based on previously observed disagreements as well as specific biological hypotheses regarding e.g. Target 2.
The experiment and analysis provided actionable results (ImageStream is considered the “real” readout”) for further work and crucial **confirmation** of conjectured problems with FACS



95% confidence intervals for estimated difference in proportion of ciliated cells between cells with given gene KO and cells with non-targeted sgRNA.
Transformed to ratio (from log10 scale).


Sureness of cilia
90-100%
80-90%
70-80%
60-70%
50-60%
40-50%
30-40%
20-30%
10-20%
0-10%
True: non-ciliated
Pred: ciliated
True: ciliated
Pred: non-ciliated

Thank you!