

INTRODUCTION

NAILS: Neurally Augmented Image Labelling Strategies

- Dr. Graham Healy, Dr. Cathal Gurrin, Dr. Tomás Ward, Prof. Alan Smeaton
- The Insight Centre for Data Analytics
- Dublin City University
- Topic: P300 signals for image labelling

OVERVIEW

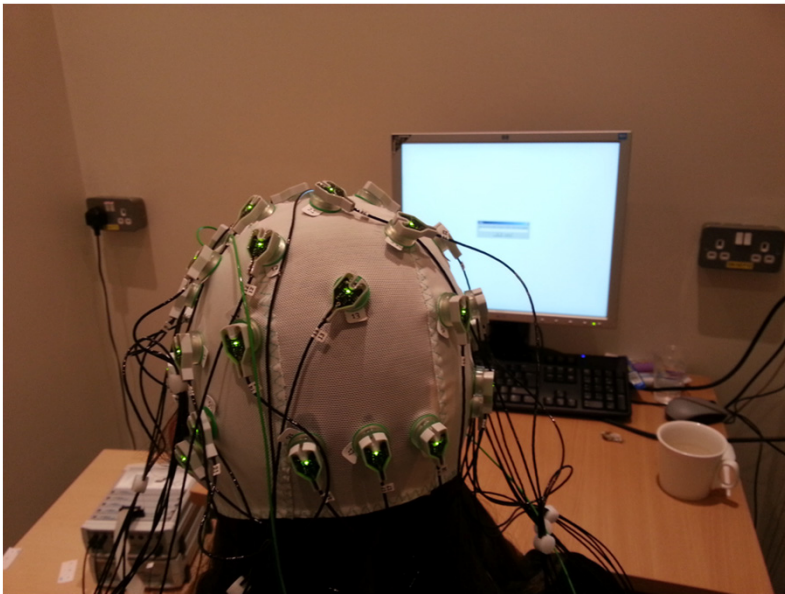
- Overview
- Examples on EEG, P300 & ERPs



- Our Task & Data Plan
- Schedule & Plan

QUICKLY: EEG

- EEG (Electroencephalography) recording systems come in many forms



P300 EXAMPLE

- This concept is best introduced with an example
- Find the image of a can of coke in the following stream of images:



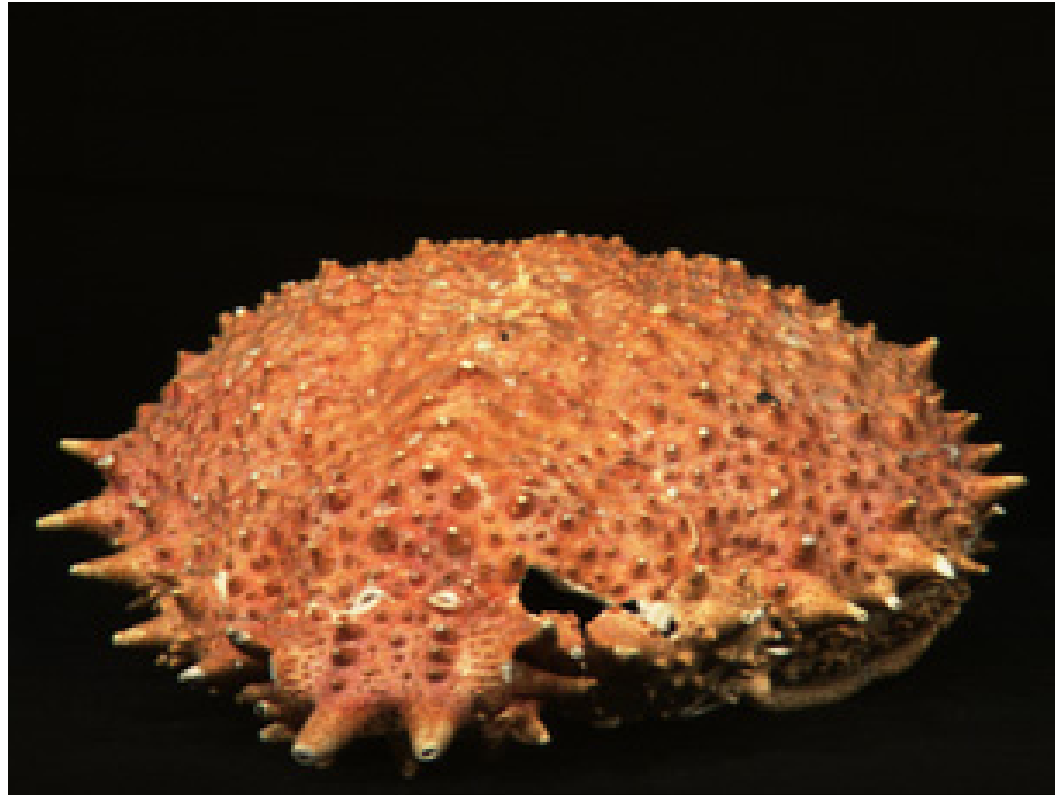
P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



P300 EXAMPLE



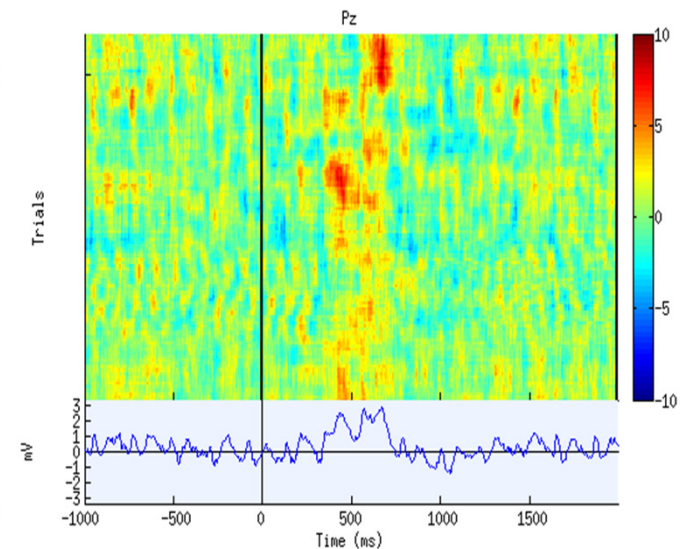
P300 EXAMPLE

- Everybody noticed it?

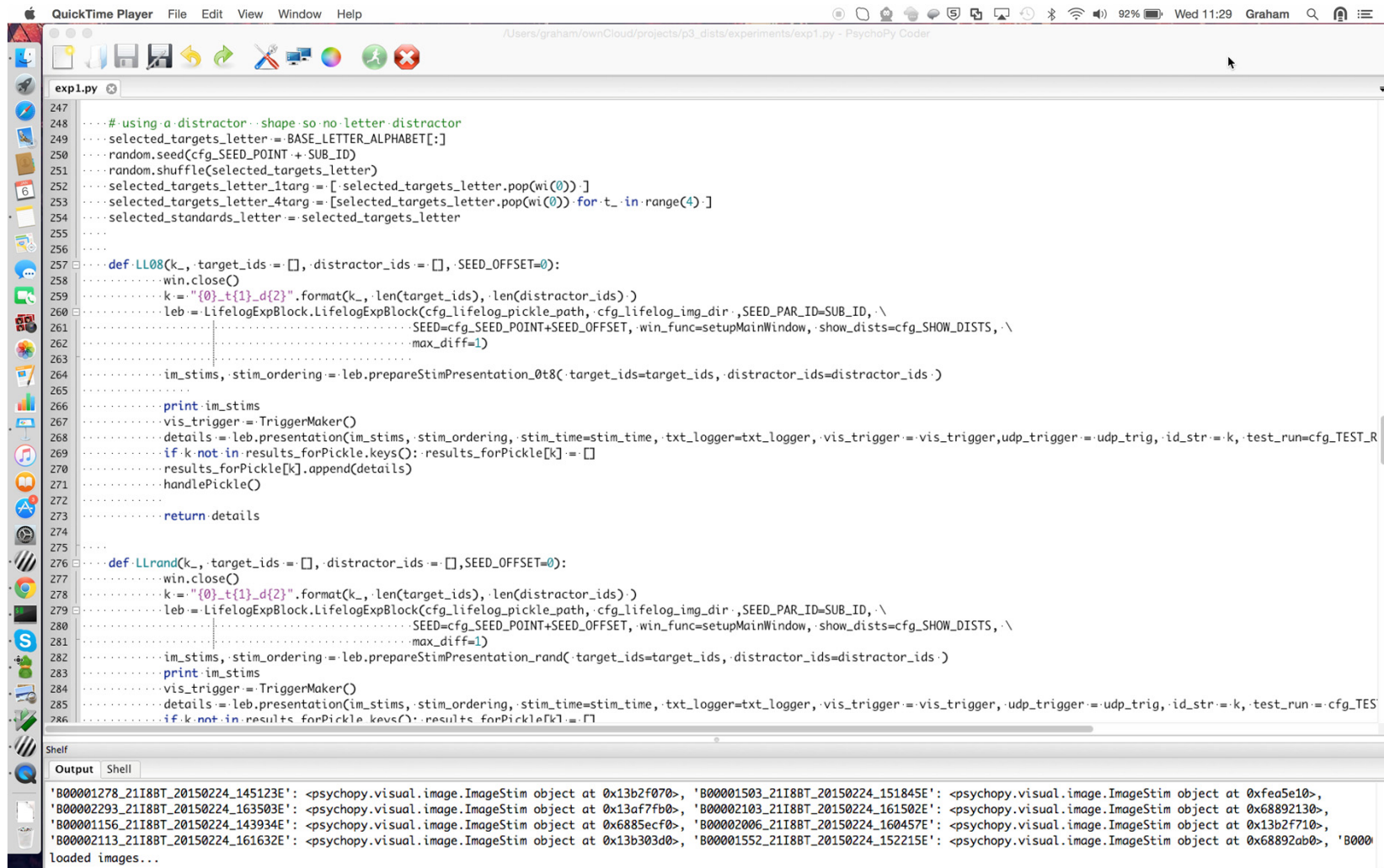


P300 RESPONSE

- There is nothing special about the can of coke, I could have told you to look out for the cup instead or used entirely different images
- This works because you do not know when the can of coke will appear and as a result when it does, your attention orientates to it (sort of like going through television stations)



ANOTHER EXAMPLE (RSVP)

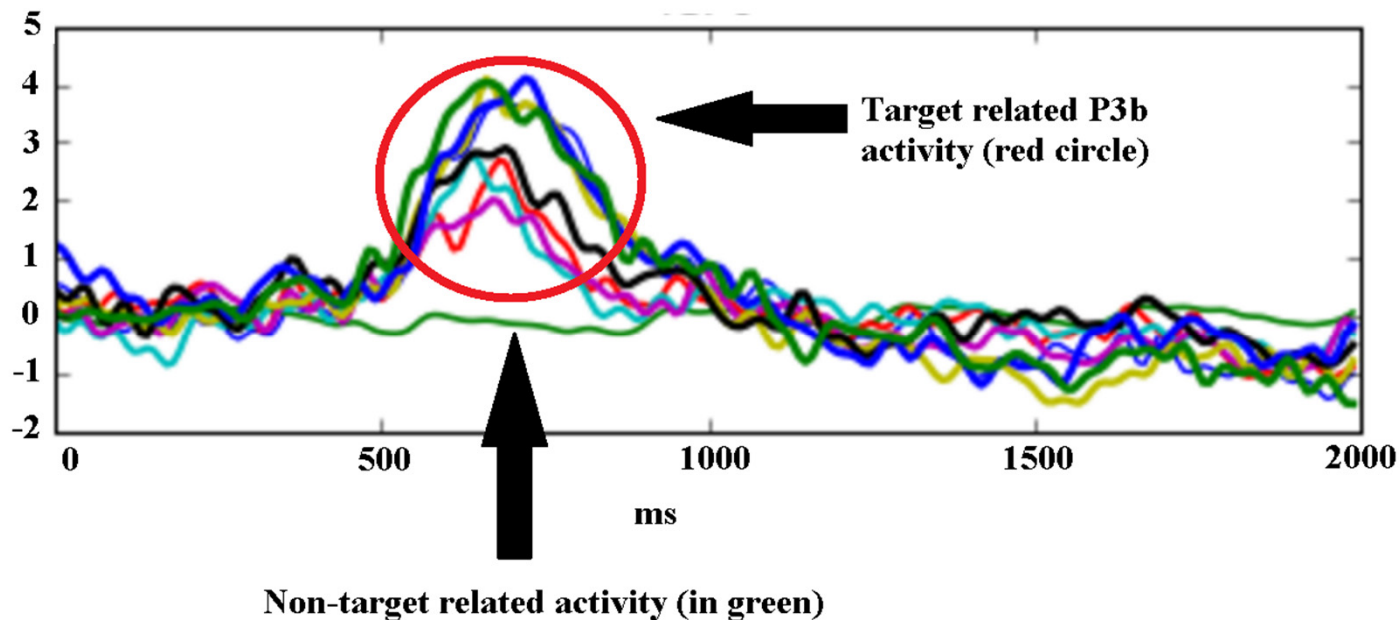


```
exp1.py
247
248 ...# using a distractor shape so no letter distractor
249 ...selected_targets_letter = BASE_LETTER_ALPHABET[:]
250 ...random.seed(cfg_SEED_POINT + SUB_ID)
251 ...random.shuffle(selected_targets_letter)
252 ...selected_targets_letter_1targ = [selected_targets_letter.pop(wi(0))]
253 ...selected_targets_letter_4targ = [selected_targets_letter.pop(wi(0)) for t in range(4)]
254 ...selected_standards_letter = selected_targets_letter
255
256
257 def LL00(k_, target_ids = [], distractor_ids = [], SEED_OFFSET=0):
258     ...win.close()
259     ...k = "{0}_{1}_{2}".format(k_, len(target_ids), len(distractor_ids))
260     ...leb = LifelogExpBlock.LifelogExpBlock(cfg_lifelog_pickle_path, cfg_lifelog_img_dir, SEED_PAR_ID=SUB_ID, \
261     ...SEED=cfg_SEED_POINT+SEED_OFFSET, win_func=setupMainWindow, show_dists=cfg_SHOW_DISTIS, \
262     ...max_diff=1)
263     ...im_stims, stim_ordering = leb.prepareStimPresentation_0t8(target_ids=target_ids, distractor_ids=distractor_ids)
264     ...print im_stims
265     ...vis_trigger = TriggerMaker()
266     ...details = leb.presentation(im_stims, stim_ordering, stim_time=stim_time, txt_logger=txt_logger, vis_trigger = vis_trigger, udp_trigger = udp_trig, id_str = k, test_run=cfg_TEST_R
267     ...if k not in results_forPickle.keys(): results_forPickle[k] = []
268     ...results_forPickle[k].append(details)
269     ...handlePickle()
270     ...return details
271
272
273
274
275
276 def LLrand(k_, target_ids = [], distractor_ids = [], SEED_OFFSET=0):
277     ...win.close()
278     ...k = "{0}_{1}_{2}".format(k_, len(target_ids), len(distractor_ids))
279     ...leb = LifelogExpBlock.LifelogExpBlock(cfg_lifelog_pickle_path, cfg_lifelog_img_dir, SEED_PAR_ID=SUB_ID, \
280     ...SEED=cfg_SEED_POINT+SEED_OFFSET, win_func=setupMainWindow, show_dists=cfg_SHOW_DISTIS, \
281     ...max_diff=1)
282     ...im_stims, stim_ordering = leb.prepareStimPresentation_rand(target_ids=target_ids, distractor_ids=distractor_ids)
283     ...print im_stims
284     ...vis_trigger = TriggerMaker()
285     ...details = leb.presentation(im_stims, stim_ordering, stim_time=stim_time, txt_logger=txt_logger, vis_trigger = vis_trigger, udp_trigger = udp_trig, id_str = k, test_run = cfg_TES
286     ...if k not in results_forPickle.keys(): results_forPickle[k] = []

Shelf
Output Shell
'B00001278_21I8BT_20150224_145123E': <psychopy.visual.image.ImageStim object at 0x13b2f070>, 'B00001503_21I8BT_20150224_151845E': <psychopy.visual.image.ImageStim object at 0xfea5e10>,
'B00002293_21I8BT_20150224_163503E': <psychopy.visual.image.ImageStim object at 0x13af7fb0>, 'B00002103_21I8BT_20150224_161502E': <psychopy.visual.image.ImageStim object at 0x68892130>,
'B00001156_21I8BT_20150224_143934E': <psychopy.visual.image.ImageStim object at 0x6885ecf0>, 'B00002006_21I8BT_20150224_160457E': <psychopy.visual.image.ImageStim object at 0x13b2f710>,
'B00002113_21I8BT_20150224_161632E': <psychopy.visual.image.ImageStim object at 0x13b303d0>, 'B00001552_21I8BT_20150224_152215E': <psychopy.visual.image.ImageStim object at 0x68892ab0>, 'B0000
loaded images...
```

P300: A LIFELOG EXAMPLE

- We've done this where participants are required to search a stream of visual data for certain targets images/concepts



* Example ICA component activity

P300-RSVP STRATEGY

- What this type of approach takes advantage of:
 - The speed of human visual recognition/processing i.e. we can often present these images at a very fast rate (example given) *via RSVP (Rapid Serial Visual Presentation)*
 - The on-the-fly capability of defining search tasks for users (within some constraints)

TASK & DATA PLAN

- For NTCIR-13 we will release a dataset from 10 experiment participants performing a number of search tasks
- The images used will be derived from the Places 2¹ dataset and the ImageNet² dataset



TASK & DATA PLAN

- The experiment for each participant will involve 8 search tasks
- i.e. *“Find photos of cats amongst the following stream of images”*
- Each search task is broken into 3 blocks (24 blocks total)
- Each block contains 450 images presented at 5 Hz
- 45 (10%) of these are targets (relevant) and 405 (90%) are non-targets(non-relevant)



TASK & DATA PLAN

- The aim for participating organisations will be to successfully predict from neural and behavioural (key press) which images are relevant (target) or non-relevant (non-target) for the relevant search query
- Participating organisations will be provided with training/validation data: 2 of 3 blocks for each search task

TASK & DATA PLAN

- During the evaluation period participating organisations will benchmark their prediction models using an online REST API (with some time-based limitations)
- We will provide raw/pre-processed feature vector for both neural / behavioural data for each image (-X task X subject) and must submit a prediction (0, 1)
- Accuracy will be assessed across all predictions for subjects X search - tasks on a withheld test set

TYPES OF SEARCH TASKS

- *Search-tasks will be modelled around those in ILSVRC
 - Four scene-centric search-tasks from Places2 i.e. find images of 'art_gallery', 'nursing_home', 'oilrig', 'television_room'
 - Four object-centric search tasks from ImageNet i.e. find images of 'cat', 'car', 'mushroom', 'desert'

· Exact search tasks will be identified during a piloting phase identifying those which a human is capable of sufficiently doing

TYPES OF DATA

- Data will be available in raw and numerous pre-processed formats to participants to lower the barrier to entry for those unfamiliar with processing EEG signals
- Neural-data feature vectors will include raw, bandpassed, wavelet, ICA, and other common feature transformations on which ML methods can be applied

TYPES OF DATA

- Example pipelines will be provided in python enabling participating organisation to leverage an already working (naive) solution to get started right away
- Before the evaluation period begins code examples will be provided showing participants how to interact with the online evaluation system

SCHEDULE & PLAN

- Next Steps:
 - Internal DCU ethics application to begin conducting experimentation (*July 2016*) – Approved (August)
 - Pilot experiments validating RT & neural responses (*Sep 2016 – Oct 2016*)
 - Collection of final experimental dataset (*Oct 2016 – Nov 2016*)
 - Dataset release (*Jan 2017*)

CONCLUSION

- We can better understand the types of search-tasks where detection of neural events are useful
- Amplify insights and perspectives on labelling-behaviour of users (i.e. nuance and state-of-mind)
- Will help to develop state-of-the-art methods for such tasks
- Will help to understand the connection between behavioural and neural responses when performing an image search-task in this way

THANK YOU

- Further information will be available as the task structure develops at:
 - ntcir-nails.computing.dcu.ie
- If any further queries please feel free to:
ghealy@computing.dcu.ie