HUMAN EVALUATION

Using Amazon Mechanical Turk
+ Google Drive

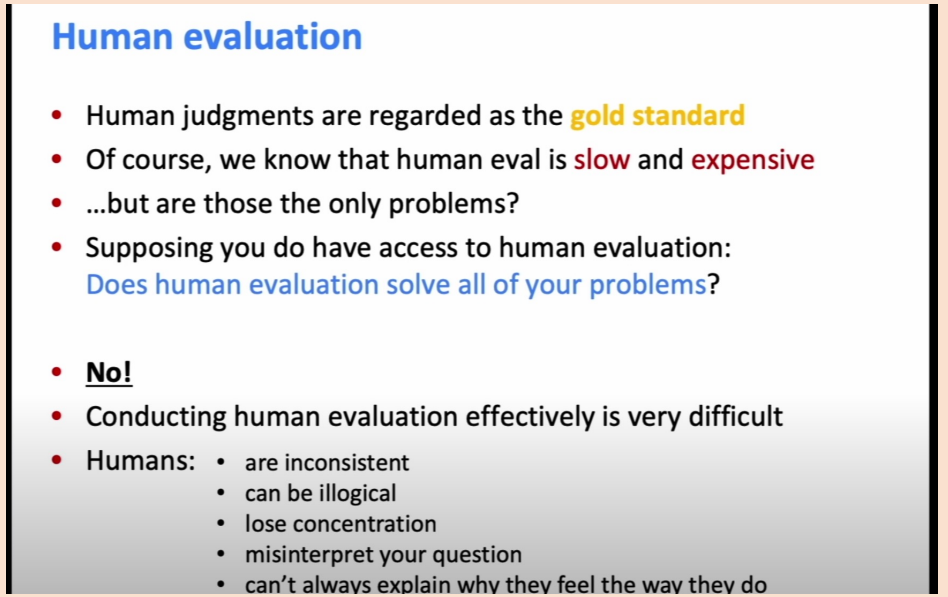
Surya

Contents

1) get account on requester.mturk.com
2) decide upon questions for survey
3) prepare html for questionnaire
4) example folder
5) prepare public urls with pydrive
6) cost / budget / no. of HITs
7) publish batch once everything is set

decide upon questions for survey

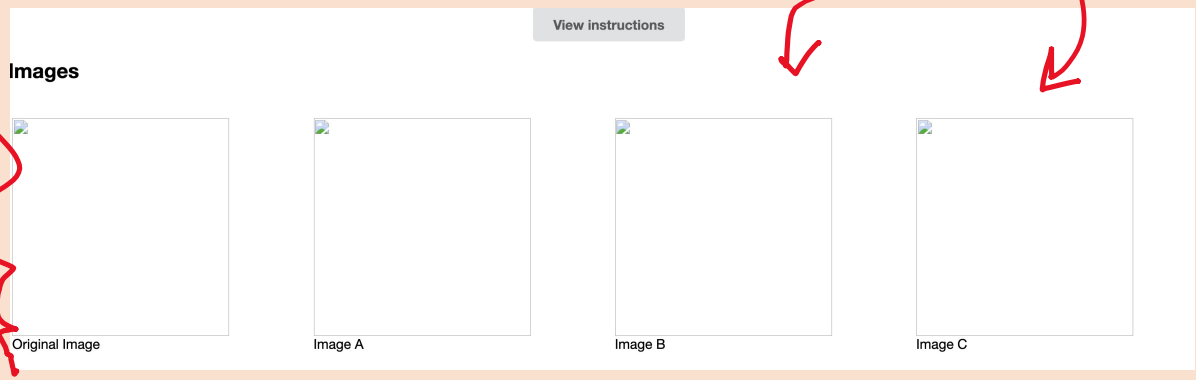
- Refer to this paper for questionnaire design https://arxiv.org/abs/1902.08654
- Related video : https://www.youtube.com/watch?v=4uG1NMKNWCU&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=15 from 1:06:13



prepare html for questionnaire

- Crowd html elements from Amazon makes it very easy to prepare UI https://docs.aws.amazon.com/sagemaker/latest/dg/sms-ui-template-reference.html
- See exp1.html / exp2.html / exp3.html
  - Replace instructions and detailed instructions as per your needs; make it comprehensive
  - What is a good annotation? What is a bad annotation?
  - Add questions
  - Add a feedback text box
  - Thank annotators for their time
- You can use css, classes, etc. all standard things from webdev; and open it in browser to see how it would look on annotator's screen

```
<h2>Images</h2>
<div class="row" style = "display:flex;">
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${orig_url}"><br/>Original Image</p>
  </div>
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${ours_url}"><br/>Image A</p>
  </div>
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${bpg_url}"><br/>Image B</p>
  </div>
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${hific_url}"><br/>Image C</p>
  </div>
</div>
```



prepare html for questionnaire

```
<h2>Images</h2>
<div class="row" style = "display:flex;">
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${orig_url}"><br/>Original Image</p>
  </div>
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${ours_url}"><br/>Image A</p>
  </div>
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${bpg_url}"><br/>Image B</p>
  </div>
  <div class="column" style = "flex: 25%; padding: 5px">
      <p><image width="256" height="256" controls source src="${hific_url}"><br/>Image C</p>
  </div>
</div>
```

Be careful with file sizes – if images are poorly reshaped, annotators will it difficult to understand

- These URLs will be populated from csv files later;
  - You can add videos similarly with <video> tag and src = $(video_url)
  - The urls are expected to be <**public links**>

	I	J	K	L
	orig_url	ours_url	bpg_url	hific_url
	https://drive.google.com/uc?id=1Dg	https://drive.google.com/uc?id=1OQng	https://drive.google.com/	https://drive.google.cor
	https://drive.google.com/uc?id=12L	https://drive.google.com/uc?id=1OKJgd	https://drive.google.com/	https://drive.google.cor

prepare html for questionnaire

Q1: Which of the three reconstructed images (A, B, C) are closer to the original image? (required)

- ○ Image A
- ○ Image B
- ○ Image C
- ○ All look close!

```
<div class="btn-group-vertical btn-block" data-toggle="buttons" id="reconstruct">
  <center>
  <h2>Q1: Which of the three reconstructed images (A, B, C) are <u>closer</u> to the original image? (required)</h2>
  <label class="btn btn-default"><input name="reconstruct" required="" type="radio" value="1" />Image A</label><br/>
  <label class="btn btn-default"> <input name="reconstruct" required="" type="radio" value="2" />Image B</label><br/>
  <label class="btn btn-default"> <input name="reconstruct" required="" type="radio" value="3" />Image C</label><br/>
  <label class="btn btn-default"> <input name="reconstruct" required="" type="radio" value="4" />All look close!</label><br/>
  </center>
</div><br/><hr/><br/>
```

These names and values help in computing the results later on from output csv – be careful while giving the names; if the value is mistakenly same for any two options – the outputs will be wrong

AO	AP	AQ	AR
Answer.reconstruct.1	Answer.reconstruct.2	Answer.reconstruct.3	Answer.reconstruct.4
TRUE	FALSE	FALSE	FALSE
TRUE	FALSE	FALSE	FALSE
TRUE	FALSE	FALSE	FALSE
TRUE	FALSE	FALSE	FALSE
FALSE	TRUE	FALSE	FALSE
FALSE	FALSE	FALSE	TRUE
TRUE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE
TRUE	FALSE	FALSE	FALSE

Response → output.csv

```
9]: def exp1_recon(df):
        opt = ["ours", "bpg", "hific", "none"]
        ans = [df[f"Answer.reconstruct.{i+1}"]
               for i in range(4)].index(True)
        return opt[ans]
```

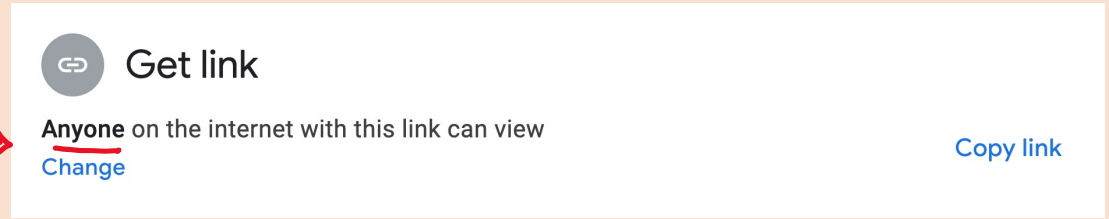
Code ← Analysis

example folder

- The following files are required for human eval experiment (questionnaire)
  - HTML file
  - Images / Videos / Text to be displayed
    - Eg: ${text_snippet_from_csv}, ${image_url}, ${video_url}, ${document_url}
    - These URLs should be public links (dropbox, gdrive, hosted on your server but anon)
    - Videos can also be private youtube / vimeo links uploaded via anon accounts
  - Input CSV file (with URLs and other details)
  - Responses CSV file (from Mturk)

prepare public urls with pydrive

- Pydrive makes it easy to iterate over files in Google drive and obtain their public urls
  - I also chose gdrive because institute provided me with 1 TB space. Typical storage is only 15 GB which may not meet your requirements
- Caveat: Pydrive makes it very hard to iterate over folders. You can only iterate over files and shortlist wrt extensions
  - Make sure that image name conveys all details about your image. Check example
- Make the image folder “shareable to all” in gdrive
  - Right click on folder; select Share
  - Only then you can move onto generating links




```
newfiles = drive.ListFile({'q': "title contains '.png' and trashed=false"}).GetList()

cnt = 0
dic = {}
for i in newfiles:
  cnt += 1
  # if cnt > 3: break
  print(i["title"])
  print(i["webContentLink"].replace("&export=download", ""))


  dic[i["title"]] = i["webContentLink"].replace("&export=download", "")
  # for k in i.keys():
    # print(k, i[k])
```

png files

not in trash

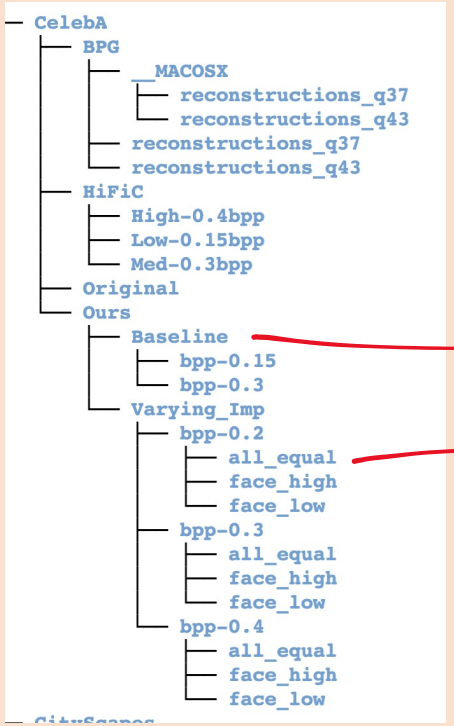
List of ALL png in gdrive

We want the image to open / be displayed in mturk rather than being downloaded; so remove the download text in url

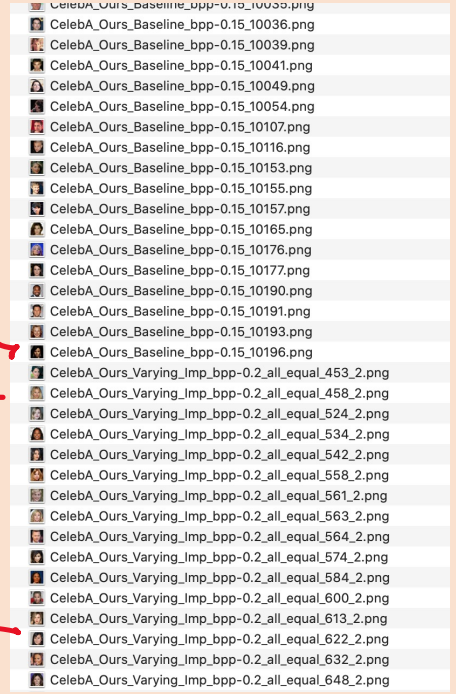
Run this on colab.research.google.com by linking the notebook with your gdrive [standard snippet in colab]; code is shared in sharepoint

prepare public urls with pydrive

- For example, CelebA baseline / model output results were arranged as follows



- I renamed and copied to a single folder as follows – carrying all folder info in the image name



Some discussion on accessing folders using code [didn't work for me]: https://stackoverflow.com/questions/40224559/list-of-file-in-a-folder-drive-api-pydrive

prepare public urls with pydrive

- Upload directory to gdrive
- Make directory shareable
- Generate public URLs for each file
- Make input csv with generated URLs
- Upload input_small.csv to requester.mturk.com and check if working [Publish Batch]


```
"https://drive.google.com/uc?id=1B9oyfGWtecjqWC0SzEF5NSb4pG7LSJAX",
"CityScapes_HiFiC_high-0.35_21.png":
"https://drive.google.com/uc?id=1SSHEKFBd6xfoL1wTy1GgktQH9Tf36pjZ",
"CelebA_HiFiC_Med-0.3bpp_10035_RECON_0.338bpp.png":
"https://drive.google.com/uc?id=1UuOIspZMRVtSwVXVwC382M7fWFK9wBJO",
"CityScapes_HiFiC_low-0.15_3.png":
"https://drive.google.com/uc?id=1YzKxiOkJuiiFAypUOHUh3qWYkObvN3I7",
"CityScapes_HiFiC_low-0.15_21.png":
"https://drive.google.com/uc?id=1s9I31BVKdw2PsJDMyw64X-LaTOA2DNhK",
"CelebA_Ours_Varying_Imp_bpp-0.2_face_low_524_0.png":
"https://drive.google.com/uc?id=1OoczBfQ-ghJHRwHvgNnWCtGiec6ntvdw",
"CityScapes_HiFiC_high-0.35_3.png":
"https://drive.google.com/uc?id=1iuAUTsP2eSOCdr5noGpjPOMUt3yIu7uq",
"CelebA_HiFiC_Low-0.15bpp_738_RECON_0.204bpp.png":
"https://drive.google.com/uc?id=1oIBKq-r-5CA5wb7Hju_rwwVK10kPp2A4",
"CityScapes_Ours_Varying_Imp_bpp-0.3_v_low_c_high_5.png":
"https://drive.google.com/uc?id=1MpyF-nW-lMpIUBVSMhCmpmzsN-IzW085",
"CelebA_HiFiC_Med-0.3bpp_613_RECON_0.373bpp.png":
```


Dict of public urls

https://pythonhosted.org/PyDrive/filelist.html

prepare public urls with pydrive

orig_image -> orig_url
hific -> hific_url
ours -> ours_url

index	orig_image	ours	hific	dataset	bpp	per	class
3	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
4	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
5	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
8	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
9	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
12	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
13	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
17	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face
3	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face
4	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face

Adding public urls to input csv

indekx	index	orig_image	ours	hific	dataset	bpp	per	class	orig_url	ours_url	hific_url
0	3	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1HwKdhyqMrmmVUrrmSSMwgo8Z43JXZpDs
1	4	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1Om2JlebsNAACfgwYbQza-TgKuJA2jd3f
2	5	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1FKSlGbQnHtEBBiFcoSfxtBYuaraE4RRO
3	8	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1MzalIm16CSypgNJd8R5W_Q3IMa4UbHZN
4	9	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1wtcEYia-G_EFybNKpSUr50gaDUnvQTVU
5	12	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1OLzegDUvEKCZbkVGTrcYoAPqvb4Zw4C9
6	13	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1-0NReLiexN3Pzt6U4Cq-GDaKgvhpTiqq
7	17	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	equal	face	https://drive	https://drive	https://drive.google.com/uc?id=1-x8y7L1fnfnCRdMaoHQOxldg9ikMrBQa
8	3	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1HwKdhyqMrmmVUrrmSSMwgo8Z43JXZpDs
9	4	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1Om2JlebsNAACfgwYbQza-TgKuJA2jd3f
10	5	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1FKSlGbQnHtEBBiFcoSfxtBYuaraE4RRO
11	8	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1MzalIm16CSypgNJd8R5W_Q3IMa4UbHZN
12	9	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1wtcEYia-G_EFybNKpSUr50gaDUnvQTVU
13	12	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1OLzegDUvEKCZbkVGTrcYoAPqvb4Zw4C9
14	13	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	high	face	https://drive	https://drive	https://drive.google.com/uc?id=1-0NReLiexN3Pzt6U4Cq-GDaKgvhpTiqq
15	3	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	low	face	https://drive	https://drive	https://drive.google.com/uc?id=1HwKdhyqMrmmVUrrmSSMwgo8Z43JXZpDs
16	4	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	low	face	https://drive	https://drive	https://drive.google.com/uc?id=1Om2JlebsNAACfgwYbQza-TgKuJA2jd3f
17	5	CelebA_Orig	CelebA_Ours	CelebA_HiFi(	CelebA	low	low	face	https://drive	https://drive	https://drive.google.com/uc?id=1FKSlGbQnHtEBBiFcoSfxtBYuaraE4RRO

cost/budget – deciding the number of HITs

- Calculate this before preparing input csv

x y z

Experiment	Type	No. of images/vids	No. of Annot	HIT Cost	Master Cost	Total cost	Total	
1	Image	51	10	0.35	0.25	0.6	306	51*10*0.6
2	Video	51	10	0.65	0.25	0.9	459	51*10*0.9
							765	

Say Budget = 500

$x_1 y_1 (z_1 + 0.25) + x_2 y_2 (z_2 + 0.25) \leq 500$

No. of experiments you want to conduct [make them disjoint]

Type of data you want the annotators to see – if it's a video, you pay them more, because greater cognitive demand

Number of annotations you want per image/video/text. Larger number => more robust responses after averaging

Total Cost = HIT Cost + Master Cost
HIT Cost = for responding to the survey
Master Cost = Mturk Masters are supposed "experts" on platform. Pay more for rich responses

The variables depend on your budget, on your algo, and other project-specific stuff

publish batch once everything is set

- Conduct a pilot study; check if annotators responses make sense
  - Read their feedback; mostly useless but some gems here and there
  - ~30 HITs sufficient
- Add sanity check questions to Accept/Reject annotator responses
  - To check if they actually read the question / checked the image, video
- Time: give annotators enough time to respond to all questions
- I'm not sure how payment is done
  - I think personal credit card; and then reimbursed at a lab level
- Paper writeup examples:
  - https://arxiv.org/abs/1902.08654 -- lots of tips in here + making a nice latex fbox
  - https://gaurav22verma.github.io/assets/papers/NonLinearConsumption.pdf