

Projet - Finding Objects In Images

Introduction

Nous retrouvons des factures dans notre quotidien, ainsi que nos bulletins de paie et relevés de compte.

C'est pour cela que, dans le cadre de notre projet, nous allons trier ces fichiers.

L'objectif est de créer un algorithme qui récupère les logos des entreprises présents sur les documents. Ainsi, nous identifierons les entreprises ayant fourni ces factures afin de les classer suivant le domaine d'activité de l'entreprise ayant réalisé la facture.

Durant ce projet, nous avons procédé à la lecture d'énormément de documentation, ainsi que des tests sur les différentes méthodes possibles de détection de logo sur les factures. Nous allons vous présenter les méthodes, algorithmes, qui ont le mieux fonctionner selon nous.

Donc, dans ce rapport, nous verrons d'abord la préparation du dataset, ensuite les méthodes utilisées pour procéder à cette classification de ces fichiers et enfin on partagera les résultats.



Préparation du datasaset

Pour pouvoir enrichir notre dataset, nous avons reçu plusieurs fichier PDF, contenant :

- Factures d'énergie (EDF, Engie, Enercorp)
- Factures de téléphonie (Free, SFR, Bouygues, Orange)
- Bulletins de paie / salaire
- Relevés de compte (LCL, Société Générale)

Ensuite, nous avons récupéré les différents logos ou références pour reconnaître un type de fichier pour pouvoir préparer notre algorithme.

Méthodes utilisées

Pour notre projet, nous avons utilisé plusieurs méthodes :

- Template Matching,
- Template Matching avec multi-scaling
- SIFT
- SIFT - Bruteforce

Nous avons aussi fait des tests sur les méthodes SURF, ORB, mahotas... mais nous allons nous concentrer uniquement sur ceux cités.

Template Matching

Cette méthode consiste à chercher et récupérer une image de modèle dans une image plus grande. Pour cela nous utilisons une fonction d'OpenCV : `cv.matchTemplate()`.

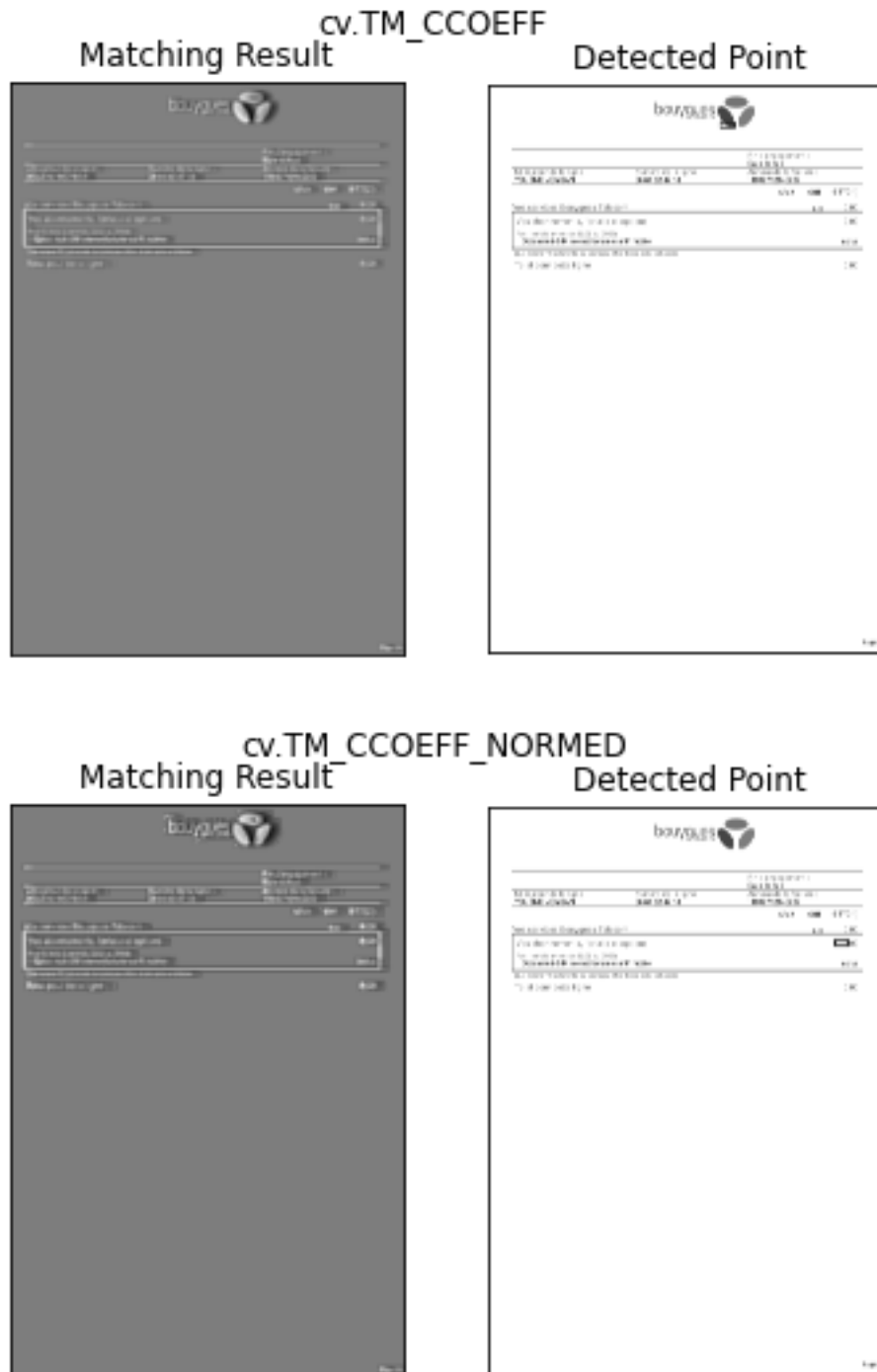
OpenCV est une bibliothèque d'open-source, permettant la vision par ordinateur, l'apprentissage automatique et le traitement d'images. En l'utilisant, on peut traiter des images pour identifier des objets, des visages ou même l'écriture manuscrite d'un humain. C'est pour cela qu'on l'utilise.

Nous commençons à importer et transformer nos PDF en format JPG. Ensuite, nous faisons un algorithme permettant la détection d'objet grâce à la fonction `matchTemplate` :

```
path=r'JPG'

img = cv.imread (path+'/factu_byt.pdf_Page 1.jpg' ,0)
img2 = img.copy()
template = cv.imread(path+'/bytt.jpg',0)
w, h = template.shape[::-1]
# Toutes les 6 méthodes de comparaison dans une liste
methods = [ 'cv.TM_CCOEFF' , 'cv.TM_CCOEFF_NORMED' , 'cv.TM_CCORR' ,
            'cv.TM_CCORR_NORMED' , 'cv.TM_SQDIFF' , 'cv.TM_SQDIFF_NORMED' ]
for meth in methods:
    img = img2.copy()
    method = eval(meth)
    # Apply template Matching
    res = cv.matchTemplate(img,template,method)
    min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)
    # if the method is TM_SQDIFF or TM_SQDIFF_NORMED, take minimum
    if method in [cv.TM_SQDIFF, cv.TM_SQDIFF_NORMED]:
        top_left = min_loc
    else:
        top_left = max_loc
    bottom_right = (top_left[0] + w, top_left[1] + h)
    cv.rectangle(img,top_left, bottom_right, (30,55,10) , 20)
    plt.subplot(121),plt.imshow(res,cmap = 'gray')
    plt.title('Matching Result'), plt.xticks([], plt.yticks([]))
    plt.subplot(122),plt.imshow(img,cmap = 'gray')
    plt.title('Detected Point'), plt.xticks([], plt.yticks([]))
    plt.suptitle(meth)
    plt.show()
    #print(bottom_right)
```

Dans ce code, on glisse l'image d'une facture ainsi que le logo qu'on recherche, ici Bouygues Télécom. Nous utilisons plusieurs méthodes déjà implémentées dans OpenCV et ainsi nous recherchons une correspondance entre les deux images. Voici ce qu'on obtient en partie :



On peut remarquer que nos méthodes détectent quelque chose, mais pas forcément au bon endroit ni correctement. La méthode qui fonctionne le mieux, c'est TM_CCOEFF

Etant donné que cette méthode n'est pas optimale et est assez aléatoire par moment, surtout sur les documents qui ne sont pas précis, nous avons testé cette méthode avec le scaling.

Template matching avec multi-scaling

La méthode template matching avec multi-scaling permet de chercher des correspondances de modèle à différentes échelles. Cela permet de détourner le problème de la dimension du logo qu'on recherche et la dimension du logo sur la facture.

Voici un exemple de détection :

The screenshot displays an Orange bill for May 29, 2020. At the top left is the Orange logo. To its right, under 'Votre offre', it lists 'Livebox Fibre' and mentions unlimited internet and TV. Further right, the bill number '01C027N748 20D9-1J05' and the date '29/05/20' are shown. Below this is a section titled 'détail de vos consommations - facture du 29.05.2020'. Underneath, it says 'total auprès d'Orange'. A table follows with columns: date, heure, intitulé, n° composé, volume, durée réelle, durée facturée, and coût. The table lists three items: a TV location for 4.99 €, a video rental for 2.00 €, and a Prada purchase for 2.99 €. A small black square icon is positioned below the table. At the bottom right, it says 'page : 3/3'.

date	heure	intitulé	n° composé	volume	durée réelle	durée facturée	coût
achats, prestations et frais ponctuels							
depuis TV							
location(s) de vidéo à la demande							4,99 €
25/05/20	21:32	Deux moi					2,00
09/05/20	20:52	Le Diable s'habille en Prada					2,99

Sur cet exemple, cela fonctionne, mais ce n'est pas toujours le cas.

L'objectif est de contrer les problèmes d'échelles en itérant la recherche du logo sur différentes tailles de l'image. Pour cela il suffit de redimensionner l'image au fur et à mesure en s'arrêtant au moment où l'image est plus petite que le logo, et garder le meilleur résultat.

Voici le code utilisé pour faire cette boucle :

```
32. | # loop over the scales of the image
33. | for scale in np.linspace(0.2, 1.0, 20)[::-1]:
34. |     # resize the image according to the scale, and keep track
35. |     # of the ratio of the resizing
36. |     resized = imutils.resize(gray, width = int(gray.shape[1] * scale))
37. |     r = gray.shape[1] / float(resized.shape[1])
38. |
39. |     # if the resized image is smaller than the template, then break
40. |     # from the loop
41. |     if resized.shape[0] < tH or resized.shape[1] < tW:
42. |         break
```

SIFT

Ensuite, nous avons fait une méthode SIFT, Scale Invariant Feature Transform. C'est un algorithme de détection de caractéristiques.

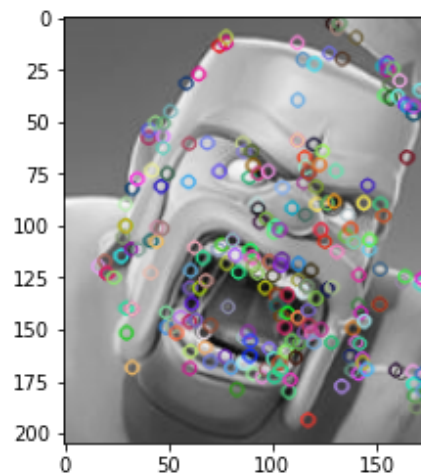
Cette méthode permet de détecter et de localiser des caractéristiques dans une image, ce qu'on appelle les « points clés ». Ainsi, ça facilite la détection.

Dans un premier temps, nous avons décidé de choisir une image lambda, assez détaillée, avant de tester sur nos factures et bulletins de salaire. Nous avons donc choisi une image de Clash Of Clan. Voici l'image d'origine :



Nous choisissons de récupérer la tête de barbare pour faire nos tests.

Voici un exemple des clés détecter, en prenant le visage du barbare :



On remarque qu'il détecte plusieurs points clés sur l'image. Maintenant, on va essayer de détecter le visage du barbare.

Voici le résultat :



On remarque que notre algorithme détecte parfaitement le visage du barbare dans l'image, grâce aux points clés. On a essayé la méthode avec des images différentes, pour pouvoir vérifier si cela fonctionnait réellement bien. On a aussi essayé avec des images différentes, qui n'avait rien à voir avec l'image à détecter d'origine, pour savoir s'il pouvait aussi détecter, mais il ne détectait pas.

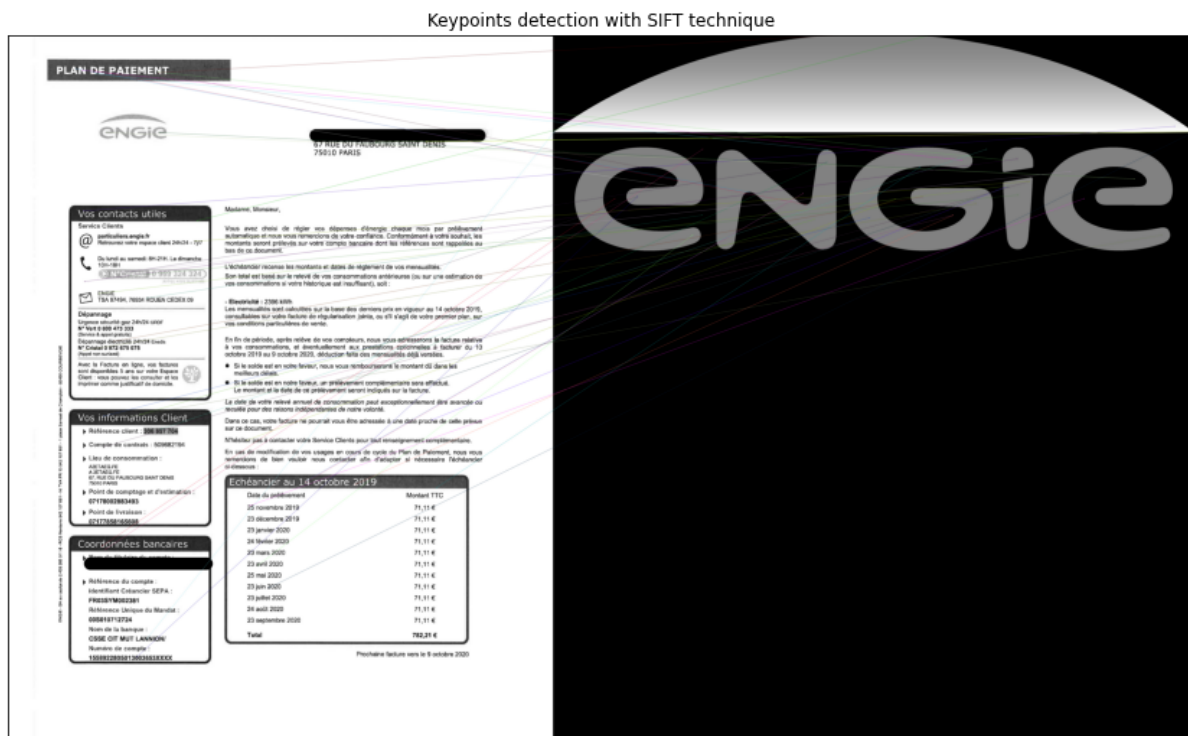
Maintenant, nous testons sur nos fichiers où on cherche à les classer.

Nous avons d'abord essayé avec des logos assez simples, où on détectait correctement. Mais, nous avons rencontré les limites de l'algorithme avec les logos avec des écritures :

Voici

le

résultat :

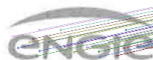


En effet, il reconnaît tous les E,N,G et I de la facture, donc cela ne permet pas la détection du logo, pourtant il est bon.

Après analyse et plusieurs tests, nous nous sommes demandé si le fait que le logo soit une image numérique et que la facture soit un scan puisse influencer sur la détection et le matching des keypoints du logo et de la facture.

Afin de vérifier notre hypothèse, nous avons donc pris des captures d'écran de l'ensemble des logos présents sur les factures. Puis, nous avons refait tourner l'algorithme.

PLAN DE PAIEMENT



67 RUE DU FAUBOURG SAINT DENIS
75010 PARIS

Vos contacts utiles

Service Clients

@ particuliers.engie.fr
Retrouvez votre espace client 24h/24 - 7j/7

☎ Du lundi au samedi: 8H-21H. Le dimanche 10H-18H

📞 N° Cristal 0 969 324 324

✉ ENGIE
TSA 87494, 76934 ROUEN CEDEX 09

Dépannage

Urgence sécurité gaz 24h/24 CRDF

N° Vert 0 800 473 333

(Service & appel gratuits)

Dépannage électricité 24h/24 Enedis

N° Cristal 0 972 675 075

(Appel non surtaxé)

Avec la Facture en ligne, vos factures sont disponibles 5 ans sur votre Espace Client : vous pouvez les consulter et les imprimer comme justificatif de domicile.

Madame, Monsieur,

Vous avez choisi de régler vos dépenses d'énergie chaque mois par prélèvement automatique et nous vous remercions de votre confiance. Conformément à votre souhait, les montants seront prélevés sur votre compte bancaire dont les références sont rappelées au bas de ce document.

L'échéancier recense les montants et dates de règlement de vos mensualités.

Son total est basé sur le relevé de vos consommations antérieures (ou sur une estimation de vos consommations si votre historique est insuffisant), soit :

- Electricité : 2396 kWh

Les mensualités sont calculées sur la base des derniers prix en vigueur au 14 octobre 2019, consultables sur votre facture de régularisation jointe, ou s'il s'agit de votre premier plan, sur vos conditions particulières de vente.

En fin de période, après relève de vos compteurs, nous vous adresserons la facture relative à vos consommations, et éventuellement aux prestations optionnelles à facturer du 10 octobre 2019 au 9 octobre 2020, déduction faite des mensualités déjà versées.

- Si le solde est en votre faveur, nous vous rembourserons le montant dû dans les meilleurs délais.
- Si le solde est en notre faveur, un prélèvement complémentaire sera effectué. Le montant et la date de ce prélèvement seront indiqués sur la facture.

La date de votre relevé annuel de consommation peut exceptionnellement être avancée ou reculée pour des raisons indépendantes de notre volonté.

Dans ce cas, votre facture ne pourrait vous être adressée à une date proche de celle prévue sur ce document.

N'hésitez pas à contacter votre Service Clients pour tout renseignement complémentaire.

En cas de modification de vos usages en cours de cycle du Plan de Paiement, nous vous remercions de bien vouloir nous contacter afin d'adapter si nécessaire l'échéancier ci-dessous :

Echéancier au 14 octobre 2019

Date du prélèvement	Montant TTC
25 novembre 2019	71,11 €
23 décembre 2019	71,11 €
23 janvier 2020	71,11 €
24 février 2020	71,11 €
23 mars 2020	71,11 €
23 avril 2020	71,11 €
25 mai 2020	71,11 €
23 juin 2020	71,11 €
23 juillet 2020	71,11 €
24 août 2020	71,11 €
23 septembre 2020	71,11 €
Total	782,21 €

Prochaine facture vers le 9 octobre 2020

Vos informations Client

▶ Référence client : 006 857 704

▶ Compte de contrats : 509682194

▶ Lieu de consommation :

A2ETAEG.FE
A2ETAEG.FE
67, RUE DU FAUBOURG SAINT DENIS
75010 PARIS

▶ Point de comptage et d'estimation :

07178002883493

▶ Point de livraison :

07177858165608

Coordonnées bancaires

▶ Nom du titulaire du compte :

▶ Référence du compte :

Identifiant Créancier SEPA :

FR03SYM002381

Référence Unique du Mandat :

005010712724

Nom de la banque :

CSSE CIT MUT LANNION/

Numéro de compte :

1558922805013003653XXXX

Comme on le voit sur l'image ci-dessus, l'algorithme détecte parfaitement le logo ENGIE. Nous avons pu observer des résultats similaires sur l'ensemble des factures Engie. Plus généralement, nous avons observé ce même résultat sur l'ensemble des factures avec les logos dits "screenshot".

Nous pouvons donc retenir de ce point que les images comparées avec SIFT doivent être du même type. Si ce n'est pas le cas, les keypoints détectés seront trop différents entre les deux images et le matching ne pourra pas se faire correctement.

L'algorithme produit des images où l'on voit le matching des keypoints comme nous avons pu le voir, pour faciliter l'analyse des résultats des algorithmes nous avons classifié et enregistré ces images dans des dossiers.

Ainsi, pour chaque facture, est créé un dossier où sont enregistrées l'ensemble des comparaisons avec les factures. De cette manière, nous avons pu plus facilement identifier le logo pour lequel la facture correspond le mieux.

SIFT - BruteForce

Nous avons utilisé une deuxième manière d'utiliser SIFT, avec SIFT BruteForce, où il prend en compte la distance des détections. La prise en compte de la distance est liée au fait que le matching ne se fait pas avec le même "matcher" que pour le SIFT précédent.

Pour la méthode bruteForce, le matcher utilisé est le "knnMatcher". Comme cela est indiqué dans le nom, ce matcher utilise des KNN. En sachant que KNN est un acronyme anglophone signifiant "K plus proches voisins". Grâce à l'implémentation du knnMatcher, nous avons pu garder seulement les keypoints respectant un certain écart de distance, en ajustant un paramètre.

Tout comme pour l'algorithme précédent nous avons enregistré les images de comparaisons en les classant dans des dossiers et par facture. Or, nous sentions que nous pourrions nous appuyer sur le paramètre de distance pour faciliter notre analyse et donc notre classification.

Nous avons donc implémenté une note pour chaque comparaison. Cette note est basée sur le nombre de keypoints détectés dans le logos, divisé par le nombre de keypoints qui ont matchés sur la facture. Cette note fait ensuite partie du nommage de l'image de comparaison.

Une fois cela effectué, nous avons continué de pousser la réflexion pour faciliter toujours un peu plus l'analyse des comparaisons et donc la classification des images. La fonction siftBruteForce retourne la note de comparaison. Nous avons donc créé un dataset recensant l'ensemble des notes pour chaque comparaison ainsi que le nom de la facture et le nom du logo qui y sont comparées.

Nous avons créé une boucle qui tourne sur chaque facture/bulletin, et sur chaque logo, comme nous pouvons le voir ci-dessous :

```
def main():

    queryImagesList = os.listdir(queryPath) #query images list
    trainImagesList = os.listdir(trainPath) #logos images list

    df_ratio_bruteforce = pd.DataFrame(columns=['Image Name', 'ratio'])

    cpt=0
    for queryImage in queryImagesList:
        print(0, " sur ", len(queryImagesList))
        for trainImage in trainImagesList:
            #Algorithm detection
            print('bruteForceMatchingSIFT : ' + queryImage + '_in_' + trainImage)
            ratio = bruteForceMatchingSIFT(queryPath, queryImage, trainPath, trainImage)
            #print('KeypointsDetectionSIFT')
            #KeypointsDetectionSIFT(queryPath+queryImage, trainPath+trainImage)

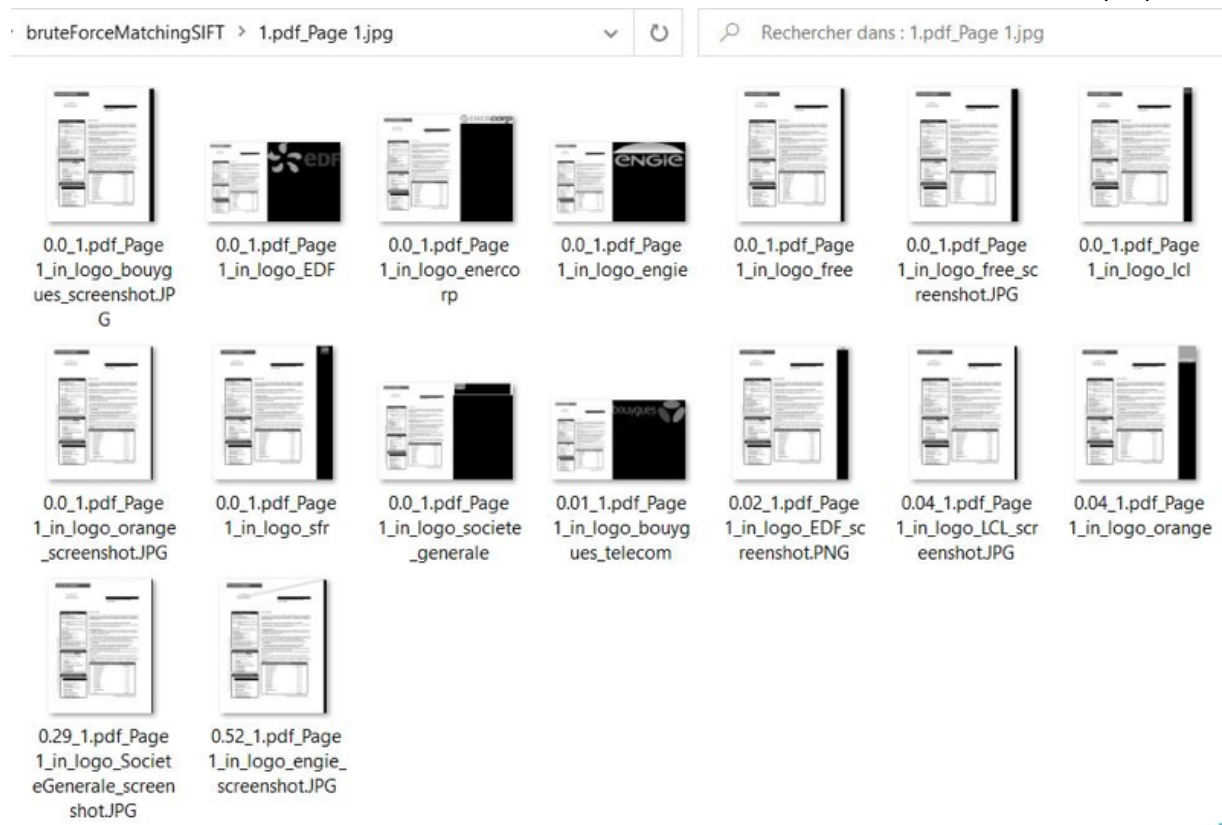
            #The dataframe to append
            data = {
                'Image Name': queryImage + '_in_' + trainImage,
                'ratio': ratio }
            df_ratio_bruteforce = df_ratio_bruteforce.append(data, ignore_index=True)

        """
        ##storing results in output.PDF:
        pdf = fpdf.FPDF(format='letter')
        pdf.add_page()
        pdf.set_font("Arial", size=12)
        pdf.write(5, "\tYolène Moysan\tAlexis Guillotin\tClément Marie-Brisson\n\n\n")
        pdf.set_font("Arial", size=20)
        pdf.write(5, "Projet Python : Finding Objects In Images\n\n\n")
        pdf.set_font("Arial", size=12)

        pdf.write(5, df_ratio_bruteforce)
        pdf.ln()
        pdf.output("output.pdf")
        """
    return df_ratio_bruteforce
df = main()
```

Voici un exemple ci-après des enregistrements des comparaisons qui sont faits dans le programme.

Nous pouvons remarquer que le nom du dossier est le nom de la facture, que chaque image est pré nommée par la note calculée par l'algorithme. Aussi, le nom de l'image contient le nom de la facture et le nom du logo comparés.



Comme on peut le voir ci-dessus, on a la note obtenue à la détection du fichier et le logo utilisé. Les meilleurs scores sont de 29% et 52%, le logo de la Société Générale et le logo ENGIE. La note la plus élevée étant celle d'Engie et la facture étant bien une facture Engie, l'algorithme a donc correctement détecté et identifié le logo.

Il est possible d'expliquer la note relativement élevée du logo de la Société Générale. En effet, cela est dû à la faible complexité du logo de cette banque. En effet, le logo de la Société Générale ne contient que très peu de détails et donc de keypoints. De cette manière, parmi le peu de keypoints détectés, beaucoup sont retrouvés dans la facture.

Comme expliqué précédemment, les notes, noms de facture et des logos ont été stockés dans un dataset par le programme Python. Puis ce dataset a été enregistré au format csv.

Voici, ci-après un extrait du dataset :

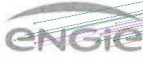
	Image Name	ratio
0	1.pdf_Page 1.jpg_in_logo_bouygues_screenshot.JPG	0.0
1	1.pdf_Page 1.jpg_in_logo_bouygues_telecom.png	0.01
2	1.pdf_Page 1.jpg_in_logo_EDF.png	0.0
3	1.pdf_Page 1.jpg_in_logo_EDF_screenshot.PNG	0.02
4	1.pdf_Page 1.jpg_in_logo_enercorp.png	0.0
5	1.pdf_Page 1.jpg_in_logo_engie.png	0.0
6	1.pdf_Page 1.jpg_in_logo_engie_screenshot.JPG	0.52
7	1.pdf_Page 1.jpg_in_logo_free.png	0.0
8	1.pdf_Page 1.jpg_in_logo_free_screenshot.JPG	0.0
9	1.pdf_Page 1.jpg_in_logo_lcl.png	0.0
10	1.pdf_Page 1.jpg_in_logo_LCL_screenshot.JPG	0.04
11	1.pdf_Page 1.jpg_in_logo_orange.png	0.04
12	1.pdf_Page 1.jpg_in_logo_orange_screenshot.JPG	0.0
13	1.pdf_Page 1.jpg_in_logo_sfr.png	0.0
14	1.pdf_Page 1.jpg_in_logo_SocieteGenerale_screenshot.JPG	0.29
15	1.pdf_Page 1.jpg_in_logo_societe_generale.png	0.0
16	13.pdf_Page 1.jpg_in_logo_bouygues_screenshot.JPG	0.0
17	13.pdf_Page 1.jpg_in_logo_bouygues_telecom.png	0.0
18	13.pdf_Page 1.jpg_in_logo_EDF.png	0.0
19	13.pdf_Page 1.jpg_in_logo_EDF_screenshot.PNG	0.15
20	13.pdf_Page 1.jpg_in_logo_enercorp.png	0.02
21	13.pdf_Page 1.jpg_in_logo_engie.png	0.0
22	13.pdf_Page 1.jpg_in_logo_engie_screenshot.JPG	0.0
23	13.pdf_Page 1.jpg_in_logo_free.png	0.0
24	13.pdf_Page 1.jpg_in_logo_free_screenshot.JPG	0.0
25	13.pdf_Page 1.jpg_in_logo_lcl.png	0.0
26	13.pdf_Page 1.jpg_in_logo_LCL_screenshot.JPG	0.04
27	13.pdf_Page 1.jpg_in_logo_orange.png	0.1
28	13.pdf_Page 1.jpg_in_logo_orange_screenshot.JPG	0.0
29	13.pdf_Page 1.jpg_in_logo_sfr.png	0.0
30	13.pdf_Page 1.jpg_in_logo_SocieteGenerale_screenshot.JPG	0.29
31	13.pdf_Page 1.jpg_in_logo_societe_generale.png	0.0
32	14.pdf_Page 1.jpg_in_logo_bouygues_screenshot.JPG	0.0

Sur cette image, nous retrouvons bien la comparaison du logo screenshot de la Société Générale à la ligne 14 avec une note de 29% et la comparaison avec le logo screenshot de Engie, à la ligne 6 avec une note de 52%.

Analyses et résultats


Voici le résultat final de la méthode SIFT Brute Force :



PLAN DE PAIEMENT




67 RUE DU FAUBOURG SAINT DENIS
75010 PARIS

Vos contacts utiles

Service Clients
 particuliers.engie.fr
 Retrouvez votre espace client 24h/24 - 7/7

 Du lundi au samedi: 8H-21H. Le dimanche 10H-18H
 **N° Cristal 0 969 324 324**
APPEL NON SURTAXÉ

 **ENGIE**
 TSA 87494, 76834 ROUEN CEDEX 09

Dépannage

Urgence sécurité gaz 24h/24 GRDF
N° Vert 0 800 473 333
 (Service & appel gratuits)
 Dépannage électricité 24h/24 Enedis
N° Cristal 0 972 675 075
 (Appel non surtaxé)

Avec la Facture en ligne, vos factures sont disponibles 5 ans sur votre Espace Client : vous pouvez les consulter et les imprimer comme justificatif de domicile.

Madame, Monsieur,

Vous avez choisi de régler vos dépenses d'énergie chaque mois par prélèvement automatique et nous vous remercions de votre confiance. Conformément à votre souhait, les montants seront prélevés sur votre compte bancaire dont les références sont rappelées au bas de ce document.

L'échéancier recense les montants et dates de règlement de vos mensualités. Son total est basé sur le relevé de vos consommations antérieures (ou sur une estimation de vos consommations si votre historique est insuffisant), soit :

- **Electricité** : 2396 kWh

Les mensualités sont calculées sur la base des derniers prix en vigueur au 14 octobre 2019, consultables sur votre facture de régularisation jointe, ou s'il s'agit de votre premier plan, sur vos conditions particulières de vente.

En fin de période, après relève de vos compteurs, nous vous adresserons la facture relative à vos consommations, et éventuellement aux prestations optionnelles à facturer du 10 octobre 2019 au 9 octobre 2020, déduction faite des mensualités déjà versées.

- Si le solde est en votre faveur, nous vous rembourserons le montant dû dans les meilleurs délais.
- Si le solde est en notre faveur, un prélèvement complémentaire sera effectué. Le montant et la date de ce prélèvement seront indiqués sur la facture.

La date de votre relevé annuel de consommation peut exceptionnellement être avancée ou reculée pour des raisons indépendantes de notre volonté.

Dans ce cas, votre facture ne pourrait vous être adressée à une date proche de celle prévue sur ce document.

N'hésitez pas à contacter votre Service Clients pour tout renseignement complémentaire.

En cas de modification de vos usages en cours de cycle du Plan de Paiement, nous vous remercions de bien vouloir nous contacter afin d'adapter si nécessaire l'échéancier ci-dessous :

Echéancier au 14 octobre 2019

Date du prélèvement	Montant TTC
25 novembre 2019	71,11 €
23 décembre 2019	71,11 €
23 janvier 2020	71,11 €
24 février 2020	71,11 €
23 mars 2020	71,11 €
23 avril 2020	71,11 €
25 mai 2020	71,11 €
23 juin 2020	71,11 €
23 juillet 2020	71,11 €
24 août 2020	71,11 €
23 septembre 2020	71,11 €
Total	782,21 €

Prochaine facture vers le 9 octobre 2020

Vos informations Client

- ▶ Référence client : **306 857 704**
- ▶ Compte de contrats : 509682194
- ▶ Lieu de consommation :
 AZETAEG.FE
 A ZETAEG.FE
 67, RUE DU FAUBOURG SAINT DENIS
 75010 PARIS
- ▶ Point de comptage et d'estimation :
07178002883493
- ▶ Point de livraison :
07177858165608

Coordonnées bancaires

▶ Nom du titulaire du compte : [REDACTED]

▶ Référence du compte :
 Identifiant Créancier SEPA :
FR03SYM002381
 Référence Unique du Mandat :
00S010712724
 Nom de la banque :
CSSE CIT MUT LANNION/
 Numéro de compte :
1558922805013003653XXXX

On remarque une très bonne détection du logo Engie, et il ne détecte pas les E/N/G/I dans le texte.

Dans les résultats, nous retrouvons nos scores dans un fichier CSV, qui nous permet de voir le ratio de chaque test de détection de logo sur un fichier.

La méthode SIFT Brute Force est la méthode qui fonctionne le mieux parmi les différents qu'on a testé. En effet, cette méthode est la plus précise et la plus adaptée à notre problématique de départ.

19/04/2022

Comme nous l'avons vu, toutes les méthodes possèdent leurs points faibles, pour SIFT c'est la distance des keypoints qui sont matchés. Nous avons pu pallier cette méthode avec SIFT - bruteforce, mais nous avons vu que bien que si le logo ne présente pas beaucoup de détails, peu de keypoints sont détectés et cela affecte la note calculée par notre programme.

Cependant, c'est bien la méthode bruteforce qui nous a permis de répondre au mieux à la problématique de départ grâce à toutes les possibilités que nous à offert le knnMatcher.