Table of Contents

1. ID MRZ RECOGNITION 2
1.1. 1. Co to jest MRZ 2
1.2. 2. Realizacja, opis program 3
1.3. 2.1 Start Training 3
1.4. 2.2 Verify Engine 5
1.5. 2.3 Photo MRZ Recognition 8
1.6. 2.3 Video MRZ Recognition 10

1.ID MRZ RECOGNITION

1.11. Co to jest MRZ

MRZ (ang. machine readable zone strefa odczytu maszynowego) to rodzaj zapisu umieszczanego na niektórych dokumentach, zawieraj cy zapisane czcionk do odczytu maszynowego (w przypadku dowodów osobistych jest to OCR-B) dane zawarte w danym dokumencie w sposób umo|liwiaj cy ich szybkie wprowadzenie do komputera. Standard zapisu jest uBatwienie komputerowego odczytu dokumentu na lotniskach. Stosowany jest w paszportach oraz niektórych dowodów osobistych uBo|one s w 3 liniach po 30 znaków. U|ywane s , wielkie litery A-Z, cyfry 0-9 oraz znak < .

Wiersz 1

Pole	Znak	Znaczenie
1	A-Z	I, A lub C na oznaczenie dokumentu podró|y innego ni| paszp
2	A-Z	typ
3-5	A-Z	kod kraju wydaj cego dokument
6-14	A-Z, 0-9	Numer dokumentu
15	0-9	cyfra kontrolna dla pól 6-14
16-30	A-Z, 0-9	Inne dane, zgodnie z uznaniem paDstwa wydaj cego

Wiersz 2

Pole	Znak	Znaczenie
1-6	0-9	Data urodzenia (RRMMDD)
7	0-9	cyfra kontrolna dla pól 1-6
8	A-Z	pBe (M dla m |czyzn, F dla kobiet, < dla pBci nieokre[lonej)
9-14	0-9	data wa|no[ci dokumentu (RRMMDD)
15	0-9	cyfra kontrolna dla pól 9-14
16-18	A-Z	obywatelstwo
19-29	A-Z, 0-9	inne (dla posiadaczy dokumentów amerykaDskich mo|e to by paszportu)
30	0-9	cyfra kontrolna dla pól 6-30 z 1. wiersza oraz 1-7, 9-15, 19-

Wiersz 3 u umieszczone s nazwisko (nazwiska) i imiona posiadacza w f W przypadku, gdy nazwisko jest dwuczBonowe, oba czBony rozdziela z to równie| s one rozdzielone znakiem < . Dla osoby, która nazywa si zapis b dzie wygl daB BACHLEDA<CURUS«ADAM<JERZY«<. Narodowe z transliterowane na odpowiednie znaki alfabetu BaciDskiego. Transliter paDstwa.

PrzykBadowy dowód osobisty	

1.22. Realizacja, opis program

Program do odczytu strefy MRZ zostaB YnTaHpOsNy(wyerwz2ju7z)ydkuem biblioteki
OpenCVjako klasyûkator w naszej sieci u|yli[my algorytmu K *NkjNNi|szych
S siadów (k-Nearest Neighbors)- (algorytm ten jest zaimplementowany

Po uruchomieniu program wy[wietli nam list zadaD które mo|na wykon



1.32.1 Start Training

Uczenie Nadzorowane - jest to funkcja, której zadaniem jest wygenero
klasyûkatorNN(pliki z tablic binarnegenerzmalkesponse,stadlitaa znaków:
generalsamples).data

Algorytm dziaBania:

Step	Description
1	Wy[wietl przygotowany przez u|ytkownika plik z danymi wzorcowymi (w zbiór znaków zapisanych przy u|yciu trzcionki ** OCR-B**. W pliku ka|d w ro|nych rozmiarach. 1.Start Training 2.Verify Engine 3.Photo MRZ Recognition 4.Video MRZ Recognition 5.Exit What would you like to do?
2	Funkcja nakBada niezb dne û ltry na plik z danymi: gray= cv2cvtCol(in, cv2COLOR_BGF2GRAY blur= cv2GaussianB(gra, (5, 5), 0) thresh= cv2adaptiveThres(blu, 25, 1, 1, 1 , 2) W wyniku czego uzyskujemy: 0 1 2 3 4 5 6 7 8 9 Q W E R T Y U I O P A S D F G H J K L Z X C V B N M < 0 1 2 3 4 5 6 7 8 9 Q W E R T Y U I O P A S D F G H J K L Z X C V B N M << 0 1 2 3 4 5 6 7 8 9 Q W E R T Y U I O P A S D F G H J K L Z X C V B N M << 0 1 2 3 4 5 6 7 8 9 Q W E R T Y U I O P A S D F G H J K L Z X C V B N M << 0 1 2 3 4 5 6 7 8 9 Q W E R T Y U I O P A S D F G H J K L Z X C V B N M <<

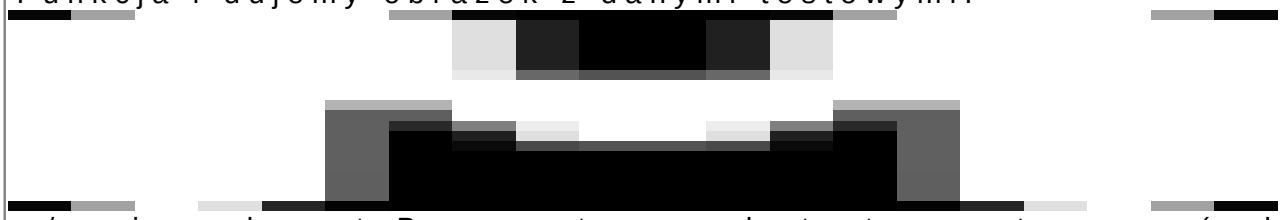
3	Nast pnie funkcja wyszukuje znaki, wycina oraz skaluje do rozmiaru 20
	imaç, contou, _ = cv2findConto(thres, cv2RETR_EXT,RNAL
	cv2CHAIN_APPROX)SIMPLE
	fo cnt in contours:
	if cv2contourA(cn) > 2:
	[x, y, w, h] = cv2boundingR(cn)
	if h > 1:
	im = cop.cop(im_co)y
	cv2.rectang(in, (x, y), (x + , y +), (0, 0, 25), 2)
	roi = thres[y:y + , x:x +]
	roismall = cv2resiz(ro, (2, 2))
	W wyniku czego otrzymujemy obraz znaku 20×20:

4	Nast pnie U|ytkownik jest proszony o wpisanie na klawiaturze jaki to z numer znaku jest zapisywany w tablicy pami ci programu.

Step	Description
5	Po klasyûkacji wszystkich znaków tablice danych s zapisane na dysku npsavetx('generalsamples.d, sampl)s npsavetx('generalresponses.d, respons)es

1.42.2 Verify Engine

Testowanie i weryûkacja nauczenia Sieci

Step	Description
1	Inicjujemy Silnik sieci oraz Badujemy dane ucz ce: sel.sampl = nploadtx("generalsamples.d, npfloat3) sel.respons = nploadtx("generalresponses., npfloat3) sel.mod = cv2m.KNearest_cr()ate sel.mod.trair(sel.sampl, cv2m.ROW_SA, sel.respons)es
2	Funkcja l dujemy obrazek z danymi testowymi.  w/w obrazek zostaB przygotowany do testu pozytywnego (zakBadam powinny by odczytane prawidBowo. Mo|na w pliku do testu zapisa zweryûkowa czy silnik nie rozpoznaje lecz w wielu Trzcionkach p zinterpretowane prawidBowo. Nasz zespóB skupiB si na porawidB wyB cznie trzcionki OCR-B (tylko ta trzcionak wyst puje w dowodac
3	NakBadamy niezb dne ûltry

4	Na podstawie konturów extraktujemy pojedyncze linie (uwaga sprav
	3 linie (obligatoryjnie dowód posiada 3) i list z parametrami poBo
	zwracamy None.
	ef extract_lin(sel, imag):
	line_list = {}
	# dilation
	cv2imsh('input extract_lin, imag)e
	kernel= cv2getStructuringEle(cv2MORPH_, (1, 1))
	img_dilatio= cv2dilat(imag, kerne, iteration=3)
	# find contours
	imag, contou, _ = cv2findConto(img_dilatiocop(),
	cv2RETR_EXTl, cv2CHAIN_APPROX)SIMPLE
	for cnt in contours:
	[x, y, v, h] = cv2boundingR(cn)
	# ic contour is big enough
	#print w * h
	if w * h> 500: # wszystko co jest poni|ej 5000 ignorujemy
	line_list[y] = [x, y, v, h]
	# sort contours
	prin "Liczba Linii % %ler(line_lis)
	if ler(line_lis) = 3:
	retur sorte(line_listitem())
	els:
	prin "Read Error, Detected Lines Count Error %-> %s"
	ler(line_lis)
	retur None

Step	Description

5	Wycinamy obrazki kaĹdej Lini na podstawie otrzymanych parametró przekazujemy je do funkcji OCR.
	`def deco(self, image):`
	`  if len(image) > 0 and image is not None:`
	`  contours = self.extract_lin(image)`
	`  MZR_lines = []`
	`  if contours is not None:`
	`  for key, value in contours:`
	`  # Get bounding box`
	`  x, y, w, h = value`
	`  line_image = image[y:y + h, x:x + w]`
	`  MZR_lines.append(self.ocr(line_image))`
	`  cv2.imshow("linie", line_image)`
	`  return MZR_lines`
	`  return None`

Step	Description

6	Funkcja OCR extraktuje z ka|dej linii znak przy u|yciu funkcji extr
	de extract_chars_from(sel, imag):
	imag, contou, _ = cv2findConto(imag, cv2RETR_EXT,RNAL
	cv2CHAIN_APPROX)SIMPLE
	char_list= []
	i = 0
	lista_rect= {}
	fo cnt in contours:
	[x, y, w, h] = cv2boundingR(cn)
	if cv2contourA(cn) > 1:
	char_list appe([x, y, w, h])
	lista_rec[x] = [x, y, w, h]
	retur sorte(lista_rec item())
	Nast pnie na podstawie zwróconych parametrów znaku funkcja OCR
	do rozmiaru 20×20 i przekazuje do Sieci KNN jako dane wej[ciowe.
	sie KNN zwraca kod rozpoznanego znaku.
	de validat(sel, read_da):a
	prin "Read Data: % read_data
	prin "Test Data: ' %sel.test_lines
	read_line_cou= le(read_da)ta
	test_line_cou = le(sel.test_line)s
	PASS_CO= 0;T
	FAIL_COU= 0;
	if (test_line_cou= read_line_co):nt
	fo i in xrang(read_line_co):nt
	read_line_dat= read_da[i]
	test_line_dat = sel.test_line[i]
	if (le(read_line_d) = le(test_line_da)):
	fo x in xrang(le(read_line_d)):a
	if read_line_d[x] = test_line_da[x]:
	PASS_COUN= 1+
	els:
	FAIL_COUN= 1+
	els:
	retur None
	els:
	retur None
	prin "Validation Summary: PASS: %s, I %(PASS_C,UNT
	FAIL_CO)JNT
	Accurac= floa(floa(PASS_C) / floa(PASS_COUNT + FAI))_COUN
	* 100
	retur Accuracy

Step	Description

7	Zwrócony String w postaci tablicy 3 linii porównujemy z wzorce. Zl
	rozpoznane i znaki zle rozpoznane na tej podstawie wyliczamy Acc
	de validat(sel, read_da):a
	prin "Read Data: % read_data
	prin "Test Data: ' %sel.test_lines
	read_line_cou= ler(read_da):ta
	test_line_cou = ler(sel.test_line)s
	PASS_CO= (;T
	FAIL_COL= (;
	if (test_line_cou= read_line_co):nt
	for i in xrang(read_line_co):nt
	read_line_dat= read_da[i]
	test_line_dat = sel.test_line[i]
	if (ler(read_line_d) = ler(test_line_da)):
	for x in xrang(ler(read_line_d)):a
	if read_line_d[›] = test_line_da[›]:
	PASS_COUN= 1+
	els:
	FAIL_COUN = 1+
	els:
	retur None
	els:
	retur None
	prin "Validation Summary: PASS: %s, I %(PASS_C,UNT
	FAIL_CO)JNT
	Accurac= floa(floa(PASS_C) / floa(PASS_COUNT + FAI))_COUN
	* 100
	retur Accuracy

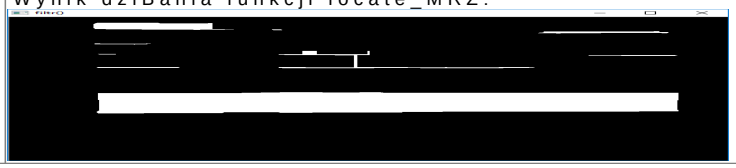
1.52.3 Photo MRZ Recognition

Zadaniem tego Modul jest zdekodowanie strefy MRZ ze zrobionego zdj

Step	Description

1	Program WB cza kamer i wy[wietla obraz, u|ytkownik robi zdj cie naciskaj c klawisz SPACE , klawisz
	Menu:
	de capture_proc(sel, ca):
	cap= cv2VideoCapt(ca)n
	whil (Tru):
	ret, frame= caprea()
	gray = cv2cvtCol(fram, cv2COLOR_BGR)2GRAY
	cv2.imsh('Camera data in, gra)
	key = cv2waitK(1)
	if ch(key &25) = ' ':
	thresh = sel.filtr(gra)
	text = sel.locate_M(thres, gra)
	prin text
	
	if ch(key &25) = 'q':
	break

Step	Description

2	Po zrobieniu zdjecia na obraz nakBadane s niezb dne û ltry a nast pnie obraz jest przekazany do funkcji
	û ltrami jest przekazany do funkcji locate_MRZ która to zlokalizuje i wekstraktuje stref MRZ z dostarczon
	strefe MRZ na dowodzie:
	`de filtr(sel, imag):`
	`rectKerne= cv2getStructuringEle(cv2MORPH_, (1, 5))`
	`sqKerne= cv2getStructuringEle(cv2MORPH_, (2 , 2))`
	`# load the image, resize it, and convert it to grayscal`
	`# smooth the image using a 3x3 Gaussian, then apply the blackhat`
	`# morphological operator to find dark regions on a light background`
	`gray= cv2GaussianB(imag, (5, 5), 0)`
	`blackha= cv2morpholog(gra, cv2MORPH_BLA, rectKern)I`
	`# compute the Scharr gradient of the blackhat image and scale the`
	`# result into the range [0, 255]`
	`grad= cv2Sob(blackh, ddep=cv2CV_3, dx=1, dy=0, ksiz=-1)`
	`grad= np.absolu(grad)X`
	`(minV, maxV) = (npmi(grad), npma(grad))`
	`grad= (25 * ((gradX - min) / (maxVal - min))).asty("uint8)`
	`# apply a closing operation using the rectangular kernel to close`
	`# gaps in between letters -- then apply Otsu's thresholding method`
	`grad= cv2morpholog(grad, cv2MORPH_, rectKern)I`
	`thresh= cv2thresho(grad, 0, 25, cv2THRESH_B | cv2THRESH_)[1]SU`
	`# perform another closing operation, this time using the square`
	`# kernel to close gaps between lines of the MRZ, then perform a`
	`# series of erosions to break apart connected components`
	`thresh= cv2morpholog(thres, cv2MORPH_, sqKern)I`
	`thresh= cv2erod(thres, Non, iteration=4)`
	`# during thresholding, it's possible that border pixels were`
	`# included in the thresholding, so let's set 5% of the left and`
	`# right borders to zero`
	`p = int(imag.shap[1] * 0.0)`
	`thres[:, :p] = 0`
	`thres[:, imag.shap[1] - p:] = 0`
	`retur thresh`
	Wynik dziBania funkc: Na miaBo zaznaczone s zlokalizowane obszary tekstu:
	
	nast pnie ten obraz przekazujemy do funkcji locate_MRZ, która rzyjmuje |e najwi kszy bialy obszar jest
	dane ze zdj cia oryginalnego (celowo zamaskowano nr dowodu) :
	`de locate_M(sel, _thres, _orygin):`
	`_orygina= cv2adaptiveThres(_orygin, 25, 1, 1, 1, 2)`
	`cnts= cv2findConto(_thres.cop(), cv2RETR_EXT , cv2CHAIN_APPROX)[-2]PLE`
	`cnts= sorte(cnt, ke=cv2contourA, revers=Tru)`
	`if le(cnt) > 0:`
	`(x, y, w, h) = cv2boundingR(cnt[0])`
	`# Shape[1] jest najwiekszy zatem to zawsze bedzie MZR`
	`pX = int((x + ) * 0.0)`
	`pY = int((y + ) * 0.0)`
	`(x, y) = (x - p , y - p )`
	`(w, h) = (w +(pX *2), h +(pY *2))`
	`result = _orygin[y:y + , x:x + ].cop()`
	`retur sel.deco(resul)`
	Wynik dziBania funkcji locate_MRZ:
	

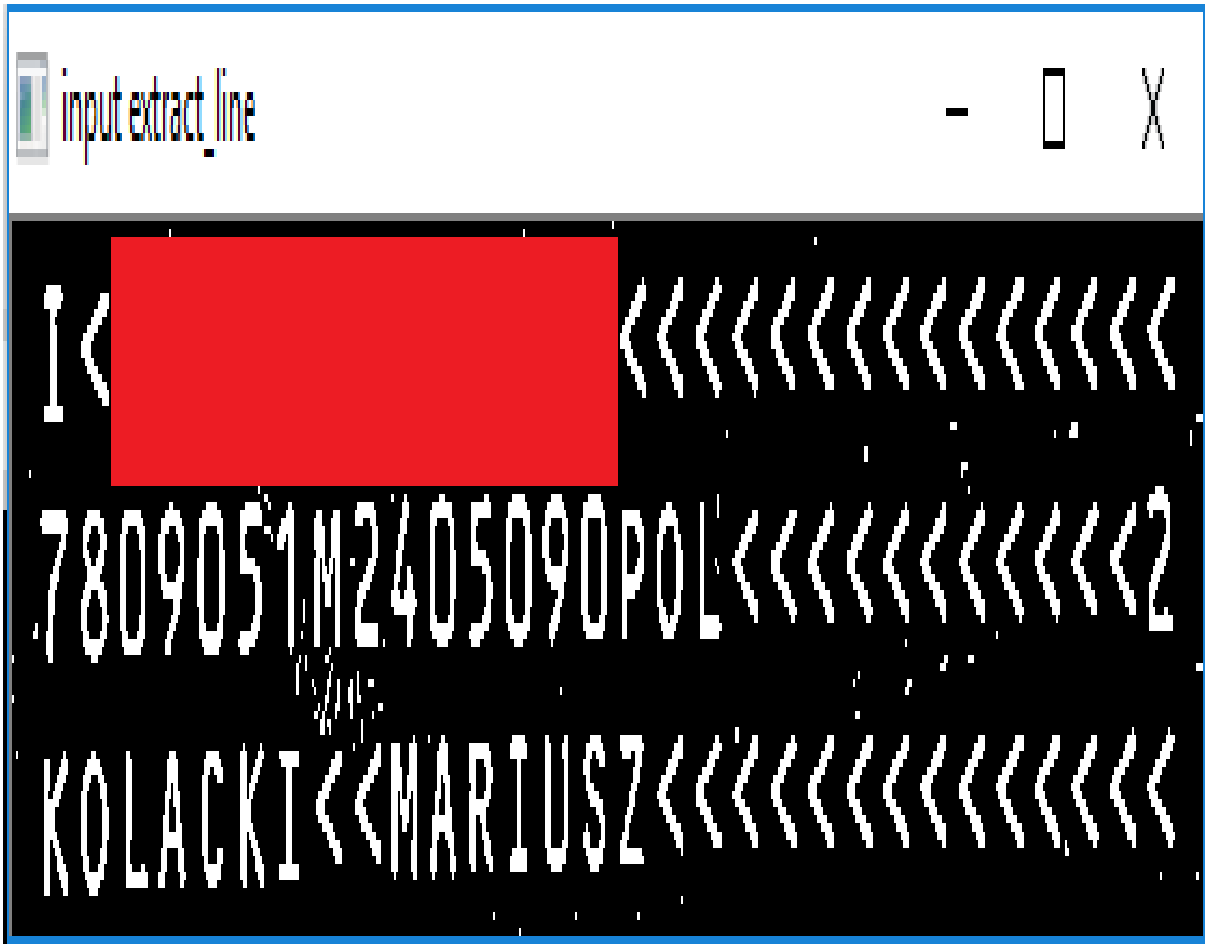
Step	Description
3	Takie zdj cie przekazane jest do funkcji Decode() która wycina obrazki ka|dej Lini na podstawie otrzyman przekazujemy je do funkcji OCR. de deco(sel, imaç): if le(imaç) > 0 an imag is no Nor e contours= sel.extract_lin(imaç)e MZR_line= [] if contour is no Nor e fo ke, value in contours: # Get bounding box x , y, v, h = value line_image= imaç[y:y + , x:x +]/ MZR_line appe (sel.oc(line_ima))e cv2.imsh("linie', line_ima)e retur MZR_lines retur None

4	Funkcja OCR extraktuje z ka|dej linii znak przy u|yciu funkcji extract_chars_from_line:
	de extract_chars_from (sel, imaç):
	imag, contou, _ = cv2findConto(imaç, cv2RETR_EXT , cv2CHAIN_APPROX)SIMPLE
	char_list= []
	i = 0
	lista_rect= {}
	fo cnt ir contours:
	[›, y, v, ł] = cv2boundingR(cn)
	if cv2contourA(cn) > 1:
	char_list appe ([›, y, v, ł])
	lista_rec[›] = [›, y, v, ł]
	retur sorte(lista_rec item())
	Nast pnie na podstawie zwróconych parametrów znaku funkcja OCR wycina obrazek, skaluje do rozmiaru wej[ciowe. Jako informacje zwrotn sie KNN zwraca kod rozpoznanego znaku.
	de oc(sel, imaç):
	sel.clear_fold('./log/ocr_imç)''
	result= ""
	chars= sel.extract_chars_from (imaç)e
	i = 0
	fo ke, value in chars:
	x , y, v, h = value
	if h > 1:
	# cv2.rectangle(oryginal,(x,y),(x+w,y+h),(0,255,0),2)
	# cv2.imshow('oryginal', oryginal)
	roi = imaç[y:y + , x:x +]/
	roismall= cv2resiz(ro, (2, 2))
	file_name= "./log/ocr_img/' % i
	sel.save_image_tc(roisma, file_nar)e
	roismall= roismal resha ((1, 40))
	roismall= npfloat3(roisma)l
	retval, result, neigh_re, dists= sel.mod.findNeare(roisma, k=1)
	strin = st(ch((result[0][0])))
	result += (strin)
	retur result

5	Wynik jest zwracany na konsol w formie tekstowej.

1.62.3 Video MRZ Recognition

Zadaniem tego Modul jest zdekodowanie strefy MRZ live na podstawi dostarczanym w czasie rzeczywistym. Funkcja ta nie wymaga naciskani automatycznie i po wykryciu 3 linie w streû e MRZ dane zostaj zdekod naniesiony na strumien danych wejsciowych. Dekodowanie odbywa si Photo MRZ Recognition pkt 2-4



From:
https://dokuwiki.zmitac.aei.polsl.pl/ - RAU6 Dokuwiki

Permanent link:
https://dokuwiki.zmitac.aei.polsl.pl/en:edu:zmitac:ai:projects:idrec[redacted]v=152

Last update: 2018/04/07 12:19