

カラーOCR ライブラリー
クラウド OCR サービス
API ユーザーズガイド
(AI 手書き文字認識)

第 4 版

不許複製

目次

1	はじめに	3
2	API	3
3	API 仕様	4
3.1	AI 手書き文字認識	4
	認識の実行.....	4
	処理状況・認識結果の取得.....	8
	データの削除.....	9
3.2	API 利用状況の確認	10
	API キー 有効期限の取得.....	10
	API キー リクエスト上限数の取得.....	11
	API キー リクエスト数の取得.....	12
4	付録	13
4.1	動作仕様	13
4.2	認識対象文字	13
	AI 手書き(英数字).....	13
	AI 手書き(住所).....	13
	AI 手書き(氏名).....	15
	AI 手書き(ふりがな).....	16
	AI 手書き(カテゴリフリー).....	16
4.3	オープンソースソフトウェアについて	19

1 はじめに

本書では、クラウド OCR サービスで提供されている AI 手書き文字認識 API (以下、「本 API」という) のインターフェースについて解説します。

2 API

すべての API アクセスは HTTPS 経由で行われ、以下の URL からアクセスされます。すべてのデータは JSON 形式として送受信されます。

API URL	説明
https://ocr.api.aia-pf.com/aitegaki/v2	・ AI 手書き文字認識の実行
https://ocr.api.aia-pf.com/aitegaki-sub/v2	・ AI 手書き文字認識の処理状況・認識結果の取得 ・ API 利用状況の確認

API キーについて

- ・ 本 API の実行には AI 手書き文字認識用の API キーが必要となります。
- ・ リクエストヘッダー「x-api-key」に有効な API キーを指定することで API を実行可能となります。

制限事項

- ・ ペイロードのサイズの上限は 10MB までとなります。
- ・ リクエストが集中した場合、処理速度低下する可能性があります。
- ・ 処理速度低下により、リクエストの実行時間が 29 秒以上経過した場合、タイムアウトとなります。タイムアウトが発生した場合を想定し、タイムアウト時のリトライ処理の実装が必要となります。

3 API仕様

3.1 AI 手書き文字認識

AI 手書き文字認識で使用する REST API です。

認識の実行

機能	AI 手書き文字認識を実行します。 ※認識結果はリクエスト ID を使用し、GET リクエストで取得します。		
URL	https://ocr.api.aia-pf.com/aitegaki/v2/fields		
Method	POST		
Content-Type	application/json		
送信 オブジェクト	名称	必須	説明
	type	○	文字認識の種別を示す値 0 : 住所 1 : 氏名 2 : カテゴリフリー (英数字、住所、氏名、ふりがな以外) 3 : 英数字 4 : ふりがな ※罫線区切りに対応しているのは 3 : 英数字、4 : ふりがなになります。
	fields	○	文字認識するフィールドの配列 ※フィールド数は最大 32 個まで指定することができます。32 個を超えて指定した場合はエラーが返却されます。 ※処理時間を最適化するためには、同じ種別の複数のフィールドをできるだけまとめて 1 回の認識実行リクエストに設定するようにしてください。
	image	○	画像ファイルのバイナリを Base64 エンコードした文字列
	category	-	認識対象とする文字種を示すオブジェクト
	symbol1	-	記号(SJIS)の有効・無効 true : 有効 (初期値) false : 無効
	symbol2	-	記号(非 SJIS)の有効・無効 true : 有効 (初期値) false : 無効
	digit	-	数字の有効・無効 true : 有効 (初期値) false : 無効
	alphabetL	-	アルファベット (大文字) の有効・無効 true : 有効 (初期値)

		false : 無効
alphabetS	-	アルファベット (小文字) の有効・無効 true : 有効 (初期値) false : 無効
hiragana	-	ひらがなの有効・無効 true : 有効 (初期値) false : 無効
katakana	-	カタカナの有効・無効 true : 有効 (初期値) false : 無効
kanji1st	-	漢字 (第 1 水準) の有効・無効 true : 有効 (初期値) false : 無効
kanji2nd	-	漢字 (第 2 水準) の有効・無効 true : 有効 (初期値) false : 無効
kanji3rd	-	漢字 (第 3 水準) の有効・無効 true : 有効 (初期値) false : 無効
kanji4th	-	漢字 (第 4 水準) の有効・無効 true : 有効 (初期値) false : 無効
kanjiExt	-	漢字 (第 1 水準～第 4 水準以外) の有効・無効 true : 有効 (初期値) false : 無効
userCharacters	-	ユーザー定義の認識対象文字 任意の文字のみを認識対象としたい場合に使用します。 ※文字認識の種別(type)ごとに規定された認識対象文字から任意の文字を指定することができます。認識対象文字以外を指定した場合は、認識結果の取得時にエラーが返却されます。
languageModel	-	言語処理の有効・無効 true : 有効 (初期値) false : 無効
multiLineFlag	-	複数行認識の指定 0 : 1 行として認識する (初期値) 1 : 複数行として認識する ※複数行認識は最初に行検出処理によって 1 行ずつに分解してから、1 行単位で文字認識を行います。処理時間を最適化するためには、複数行として認識する必要がある場合に限って、

			<p>複数行認識を指定してください。</p> <p>※1 回の認識実行リクエスト内に、本パラメーターに異なる値を混在して設定することはできません。混在して設定した場合は、認識結果の取得時にエラーが返却されます。</p>
	respectVsMarks	-	<p>濁点・半濁点を 1 文字として認識</p> <p>0 : 濁点・半濁点を 1 文字として認識しない (初期値)</p> <p>1 : 濁点・半濁点を 1 文字として認識する</p> <p>type にて文字認識の種別「4 : ふりがな」を指定した際に有効となるパラメーターとなります。濁点・半濁点を 1 文字として認識するかどうかを指定します。</p> <p>【例】 パナソニック</p> <p>認識結果</p> <p>0 を設定した場合 : パナソニック</p> <p>1 を設定した場合 : ハ° ナソニック</p> <p>※0 を設定したとき、濁点・半濁点を直前の文字と 1 文字として結合した場合に、結合後の文字が認識対象文字 (文字コード) にならない場合は、当該濁点・半濁点は認識結果から削除されます。</p> <p>【例】 うゝ いんす ⇒ ういんす</p> <p>※「うゝ」は[AI 手書き (ふりがな)]の認識対象文字ではないため。認識対象文字については、「4.1 動作仕様」の「AI 手書き (ふりがな)」をご参照ください。</p>
受信 オブジェクト	名称	説明	
	requestId	リクエスト ID 認識結果の取得に使用します。	
備考	<ul style="list-style-type: none"> 上記 URL に POST でリクエストし、リクエスト Body に JSON 文字列を設定して送信してください。 本 POST でアップロードしたデータと認識処理結果データの保持期間は、本 POST リクエストの受信時から 20 分間です。本 POST リクエストを送信して 1 秒程度後に GET リクエストによる認識結果取得を開始してください。保持期間を経過したデータは自動的に削除され、削除後は当該データの認識結果は取得できません。 サーバーが混み合っている場合、上記保持期間中に認識処理が完了せず、GET リクエストによる認識結果取得が上記保持期間中に成功しない場合があります。その場合は、時間を空けて再度 POST リクエストからやり直すようなリトライ処理の実装が考えられます。 		
送信 JSON サンプル	<pre>{ "type":0 "fields":[{ "image":"(base64 エンコードした画像のデータ)", "category":{" "symbol1":true,</pre>		

	<pre> "symbol2":true, "digit":true, "alphabetL":true, "alphabetS":true, "hiragana":true, "katakana":true, "kanji1st":true, "kanji2nd":true, "kanji3rd":true, "kanji4th":true, "kanjiExt":true }, "userCharacter":"(ユーザー定義文字)", "languageModel":true "multiLineFlag":1 }, { "image":"(base64 エンコードした画像のデータ)", ... }] } </pre>
<p>受信 JSON サンプル</p>	<pre> {"requestId":"(リクエスト ID)"} </pre>

処理状況・認識結果の取得

機能	AI 手書き文字認識処理の処理状況、および認識結果を取得します。	
URL	https://ocr.api.aia-pf.com/aitegaki-sub/v2/fields/{requestId}	
Method	GET	
パス パラメーター	requestId : AI 手書き文字認識 (POST) で返却されたリクエスト ID を指定します。	
受信 オブジェクト	名称	説明
	requestId	リクエスト ID
	status	処理状況を示す文字列 RUNNING : 処理中 SUCCESS : 文字認識処理成功 FAILURE : 文字認識処理中にエラーが発生した場合
	fields	フィールドオブジェクトの配列
	text	認識結果文字列
	score	確度を示す数値
	error	エラー情報を格納するオブジェクト
	errorCode	エラーコード
	message	エラーメッセージ
備考	<ul style="list-style-type: none"> 上記 URL にリクエスト ID を指定し、GET でリクエストしてください。 成功した場合は処理状況、および認識結果を返却します。 処理状況(status)が RUNNING の場合は、処理状況(status)が SUCCESS もしくは FAILURE となるまで、本 GET リクエストを、ある程度の時間間隔 (1 秒など) を空けながら、繰り返し送信するのが一般的な実装となります。 	
受信 JSON サンプル	<pre>{ "requestId": "(リクエスト ID)", "status": "SUCCESS", "fields": [{ "text": "(認識結果文字列)", "score": 97.3863144 }, ...], "error": null }</pre>	

データの削除

機能	AI 手書き文字認識処理でシステム上に作成されたデータを削除します。 ※現在、システム上に作成されたデータは認識処理完了後に即時削除されているので、本 API によって明示的に削除する必要はありません。	
URL	https://ocr.api.aia-pf.com/aitegaki-sub/v2/fields/{requestId}	
Method	DELETE	
パス パラメーター	requestId : AI 手書き文字認識 (POST) で返却されたリクエスト ID を指定します。	
受信 オブジェクト	名称	説明
	requestId	削除したデータのリクエスト ID
備考	<ul style="list-style-type: none"> ・上記 URL にリクエスト ID を指定し、DELETE でリクエストしてください。 ・成功した場合はリクエスト ID を返します。 	
受信 JSON サンプル	{"requestId":"(リクエスト ID)"}	

3.2 API 利用状況の確認

AI 手書き文字認で指定する API キーの利用状況の確認を行う REST API です。

API キー 有効期限の取得

機能	API キーに設定されている有効期限を取得します。				
URL	https://ocr.api.aia-pf.com/aitegaki-sub/v2/tags/expiry-date				
Method	GET				
Content-Type	application/json				
受信 オブジェクト	名称	説明			
	expiryDate	有効期限 (yyyy/MM/dd) ※API キーに有効期限が設定されていない場合は空文字列が返却されます。			
	error	エラー情報を格納するオブジェクト			
	<table border="1"> <tr> <td>errorCode</td> <td>エラーコード</td> </tr> <tr> <td>message</td> <td>エラーメッセージ</td> </tr> </table>	errorCode	エラーコード	message	エラーメッセージ
errorCode	エラーコード				
message	エラーメッセージ				
備考	<ul style="list-style-type: none"> 無効な API キーが指定された場合は、レスポンスエラーとなります。 成功した場合は、API キーの有効期限を返却します。 				
受信 JSON サンプル	<pre>{ "expiryDate" : " (日付) ", "error" : null }</pre>				

API キー リクエスト上限数の取得

機能	API キーに設定されている現在のリクエスト上限数/月を取得します。 ※取得できるのは現在設定されているリクエスト上限数のみとなり、過去設定されていたリクエスト上限数を取得することはできません。	
URL	https://ocr.api.aia-pf.com/aitegaki-sub/v2/tags/max-requests-per-month	
Method	GET	
Content-Type	application/json	
受信 オブジェクト	名称	説明
	maxRequestsPerMonth	リクエスト上限数/月 ※API キーにリクエスト上限数が設定されていない場合は空文字列が返却されます。
	error	エラー情報を格納するオブジェクト
	errorCode	エラーコード
	message	エラーメッセージ
備考	<ul style="list-style-type: none"> 無効な API キーが指定された場合は、レスポンスエラーとなります。 成功した場合は、設定されている現在のリクエスト上限数/月を返却します。 	
受信 JSON サンプル	<pre>{ "maxRequestsPerMonth" : " (リクエスト上限数) ", "error" : null }</pre>	

API キー リクエスト数の取得

機能	<p>指定された年月の API キーのリクエスト数を取得します。</p> <p>※指定された年月は、UTC (協定世界時) 基準となります。</p> <p>例) 2020 年 7 月を指定してリクエスト数を取得した場合、以下の期間のリクエスト数が取得されます。</p> <p>UTC(協定世界時) : 2020 年 7 月 1 日 0:00:00 ~ 2020 年 7 月 31 日 24:00:00</p> <p>JST(日本標準時) : 2020 年 7 月 1 日 9:00:00 ~ 2020 年 8 月 1 日 9:00:00</p> <p>※以下の年月を指定しての API 呼び出しは無効です。</p> <ul style="list-style-type: none"> ・ 現在年月より未来の年月 ・ API キーが発行されていない過去の年月 											
URL	<p><code>https://ocr.api.aia-pf.com/aitegaki-sub/v2/usage?year={year}&month={month}</code></p>											
Method	<p>GET</p>											
クエリ パラメーター	<p>{year} : 年 (西暦)</p> <p>{month} : 月</p> <p>※年月は UTC(協定世界時)として指定したことになります。</p>											
Content-Type	<p>application/json</p>											
受信 オブジェクト	<table border="1"> <thead> <tr> <th>名称</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>usedQuota</td> <td>リクエスト数</td> </tr> <tr> <td>error</td> <td>エラー情報を格納するオブジェクト</td> </tr> <tr> <td> errorCode</td> <td>エラーコード</td> </tr> <tr> <td> message</td> <td>エラーメッセージ</td> </tr> </tbody> </table>	名称	説明	usedQuota	リクエスト数	error	エラー情報を格納するオブジェクト	errorCode	エラーコード	message	エラーメッセージ	
名称	説明											
usedQuota	リクエスト数											
error	エラー情報を格納するオブジェクト											
errorCode	エラーコード											
message	エラーメッセージ											
備考	<ul style="list-style-type: none"> ・ 無効な API キーが指定された場合は、レスポンスエラーとなります。 ・ 成功した場合は、指定された年月の API キーのリクエスト数を返却します。 											
受信 JSON サンプル	<pre>{ "usedQuota" : " (リクエスト数) ", "error" : null }</pre>											

4 付録

4.1 動作仕様

入力画像	スキャナー解像度	50~2,400dpi (400dpiを推奨)	
	画像サイズ	画像の長辺が2,400px以下を推奨	
	対応形式	BMP, GIF, JPEG, PNG, TIFF (※1) ※ただし、.NET FrameworkのSystem.Drawing.Bitmapクラスがサポートする画像形式に限る。 ※1 マルチページには非対応	
日本語文字認識	対象文字	英数字	英字, 数字, 記号の一部
		住所	英字, 数字, ひらがなの一部, カタカナ, ギリシャ文字の一部, 記号の一部, 第1水準漢字の一部(2,015字), 第2水準漢字の一部(366字)
		氏名	ひらがなの一部, カタカナの一部, 記号の一部, 第1水準漢字の一部(2,368字), 第2水準漢字の一部(466字), 第3水準漢字の一部(8字), JIS規格外漢字(1字)
		ふりがな	ひらがな, カタカナ, 記号の一部(6字)
		カテゴリフリー	英字, 数字, ひらがな, カタカナ, ギリシャ文字の一部, 記号の一部, 第1水準漢字の一部(2,965字), 第2水準漢字の一部(1,210字), 第3水準漢字の一部(24字), 第4水準漢字の一部(2字), JIS規格外漢字の一部(3字)

4.2 認識対象文字

AI 手書き文字認識における認識対象文字の一覧です。

種別 (英数字、住所、氏名、カテゴリフリー、ふりがな) ごとに認識対象文字の範囲が異なります。

AI 手書き (英数字)

英字 (52 字)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z
数字 (10 字)
0 1 2 3 4 5 6 7 8 9
JIS 記号(一部) (24 字)
" # \$ % & ' () * +, -. / : < = > ? @ [¥] _

AI 手書き (住所)

英字 (52 字)
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z
数字 (10 字)
0 1 2 3 4 5 6 7 8 9
ひらがな(一部) (78 字)
あ あ い う え お か が き ぎ く ぐ け げ こ さ ざ し じ す ず せ ぜ そ ぞ た だ ち つ づ て で と ど な に ぬ ね の は ば ぱ ひ び び ふ ぶ へ べ へ ぼ ぼ ま み む め も や や ゆ ゅ よ り り れ ろ わ ゑ ん
カタカナ (86 字)
ア アイ ウ ウ ヲ ヲ エ エ オ オ カ カ ガ キ グ グ ケ ケ ゴ ゴ サ サ ジ ジ ス ス セ セ ソ ソ タ タ チ チ ヅ ヅ ツ ツ テ テ ト ト ナ ナ ニ ニ ネ ノ ハ バ パ ヒ ビ ピ フ ブ ヘ ベ ペ ホ ボ マ ミ ム メ モ ヤ ヨ ユ ヨ ヲ ヲ ラ リ ル レ ロ ヲ ヲ キ エ ヲ ン
ギリシャ文字(一部) (1 字)
α

4.3 オープンソースソフトウェアについて

AI 手書き文字認識には、次のオープンソースソフトウェアが含まれます。

ソフトウェア名称	ライセンス
OpenCV	Copyright (C) 2000–2015, Intel Corporation, all rights reserved. Copyright (C) 2009–2011, Willow Garage Inc., all rights reserved. Copyright (C) 2009–2015, NVIDIA Corporation, all rights reserved. Copyright (C) 2010–2013, Advanced Micro Devices, Inc., all rights reserved. Copyright (C) 2015, OpenCV Foundation, all rights reserved. Copyright (C) 2015, Itseez Inc., all rights reserved.
Tensorflow	Copyright 2018 The TensorFlow Authors. All rights reserved.