

# Amazon Rekognition 試してみた

<ENOG74 Meeting>

2022/06/10

Yasuyuki Kaneko  
yasuyuki.kaneko@global-netcore.jp

はじめに . . .

- Network Operators' Groupなのに、全然ネットワークな話じゃなくて、ごめんなさい。。。



# Amazon Rekognitionとは

- **機械学習の専門知識を必要とせず**、気軽に画像認識や映像分析を行うことができる、AWSのサービスです
- 顔の検出と分析、同一人物判定、有名人の認識、不適切コンテンツの検出、文字検出などができます
- 詳しくはこちらを・・・
  - <https://aws.amazon.com/jp/rekognition/>

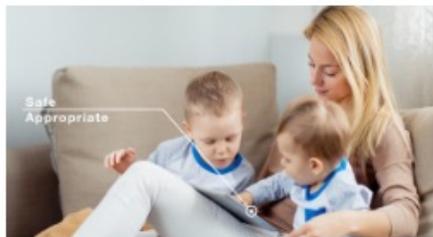


# Amazon Rekognitionで

こんなことができますよ！

今回お試ししてみたのは、

「顔比較」です



## コンテンツのモデレーション

画像と動画全体で潜在的に危険、不適切、または不要なコンテンツを検出します。

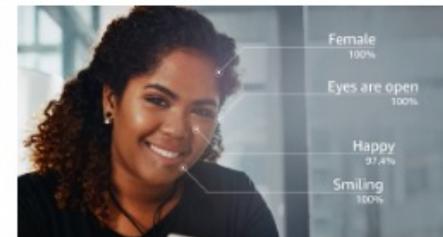
[詳細はこちら](#)



## 顔比較と検索

別の写真またはプライベートな画像リポジトリからの顔の類似性を判断します。

[詳細はこちら](#)



## 顔検出と分析

画像や動画に映っている顔を検出して、それぞれの開いている目、眼鏡、顔の毛などの属性を認識します。

[詳細はこちら](#)



## ラベル

何千ものオブジェクトとシーンに加えてアクティビティを検出します。たとえば「パッケージの配信」や「サッカーの試合」など。

[詳細はこちら](#)



## カスタムラベル

自動機械学習 (AutoML) を使用してブランドロゴなどのカスタムオブジェクトを検出し、わずか 10 画像でモデルをトレーニングします。

[詳細はこちら](#)



## テキスト検出

道路標識、ソーシャルメディア投稿、および製品パッケージの画像や動画から歪んだり曲がったりしているテキストを抽出します。

[詳細はこちら](#)



## 有名人の認識

メディア、マーケティング、および広告用の写真や映像をカタログに載せる有名人を特定します。

[詳細はこちら](#)



## 動画セグメント検出

ブラックフレーム、開始クレジットまたは終了クレジット、スレート、カラーバー、ショットなど、動画の主要なセグメントを検出します。

[詳細はこちら](#)

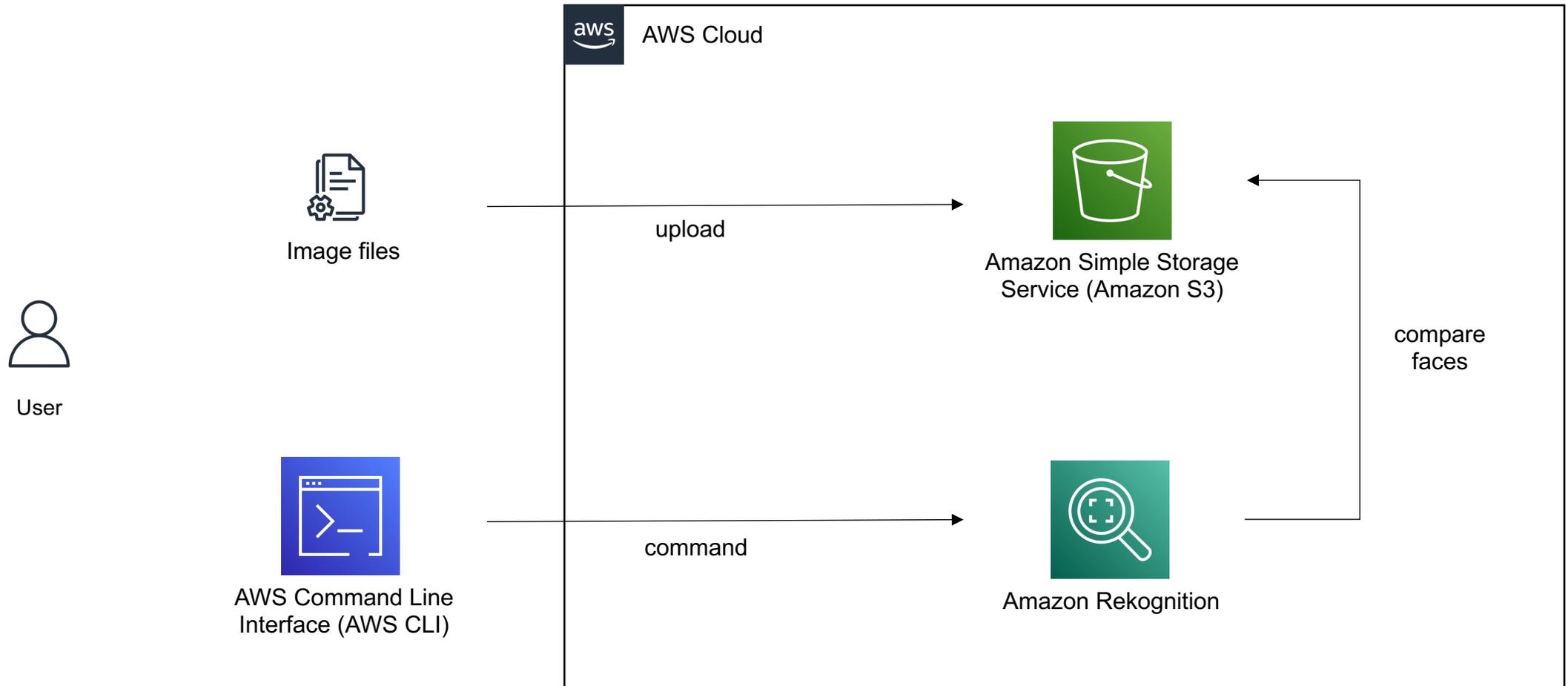


## Streaming Video Events の検出

ライブビデオストリームから、パッケージ、ペクト、人などのオブジェクトをリアルタイムで検出します。

[詳細はこちら](#)

# テスト構成



# やり方

- S3にbucketを作成
- S3 bucketに画像データをアップロード
- AWS CLIでRekognitionを呼び出し、結果を得る

```
$ aws rekognition compare-faces ¥  
  --source-image '{"S3Object" : {"Bucket": "pictures", "Name": "face1.jpg"}}' ¥  
  --target-image '{"S3Object" : {"Bucket": "pictures", "Name": "face2.jpg"}}'
```

# 結果 (json形式の場合)

```
{
  "SourceImageFace": {
    "BoundingBox": {
      "Width": 0.21217675507068634,
      "Height": 0.23572778701782227,
      "Left": 0.3921152353286743,
      "Top": 0.09790350496768951
    },
    "Confidence": 99.9991455078125
  },
  "FaceMatches": [
    {
      "Similarity": 99.88639068603516,
      "Face": {
        "BoundingBox": {
          "Width": 0.2455182522535324,
          "Height": 0.2682085633277893,
          "Left": 0.5401257276535034,
          "Top": 0.3119314908981323
        },
        "Confidence": 99.99772644042969,
        "Landmarks": [
          {
            "Type": "eyeLeft",
            "X": 0.7026223540306091,
            "Y": 0.42617806792259216
          },
          {
            "Type": "eyeRight",
            "X": 0.6797612905502319,
            "Y": 0.5291976928710938
          }
        ]
      }
    }
  ]
}
```

```
{
  "Type": "mouthLeft",
  "X": 0.6106165051460266,
  "Y": 0.401437371969223
},
{
  "Type": "mouthRight",
  "X": 0.5919978022575378,
  "Y": 0.4875119626522064
},
{
  "Type": "nose",
  "X": 0.6357628107070923,
  "Y": 0.4850003719329834
}
],
"Pose": {
  "Roll": 105.6818618774414,
  "Yaw": 17.30537986755371,
  "Pitch": 2.4946892261505127
},
"Quality": {
  "Brightness": 82.38579559326172,
  "Sharpness": 92.22801208496094
}
},
"UnmatchedFaces": []
}
```

# 見るべきポイントは

```
"FaceMatches": [  
  {  
    "Similarity": 99.88639068603516,  
    //中略//  
  }  
],  
"UnmatchedFaces": []
```

一致

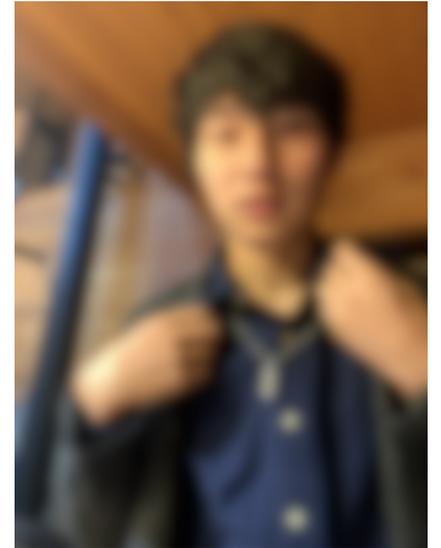
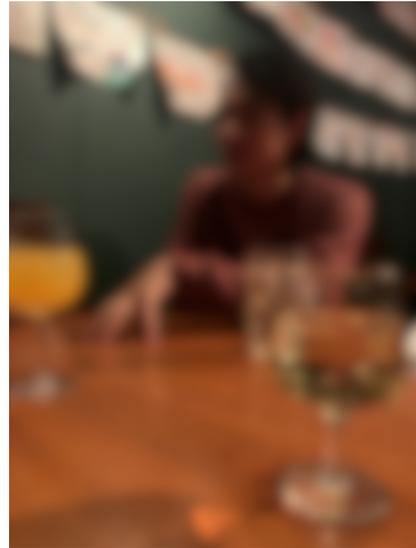
類似度スコア

```
"FaceMatches": [],  
"UnmatchedFaces": [  
  {  
    //中略//  
  }  
]
```

不一致

# テストデータ：長男

2002年生まれ 大学生

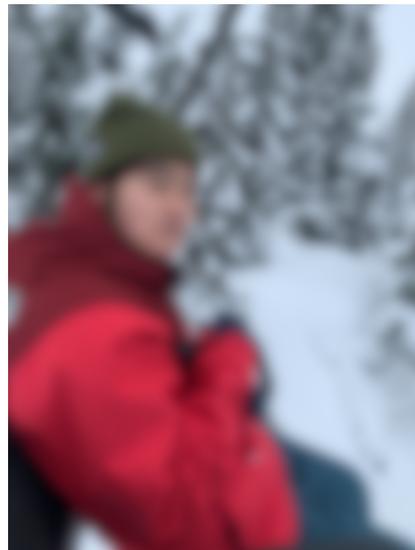


# テストデータ：次男

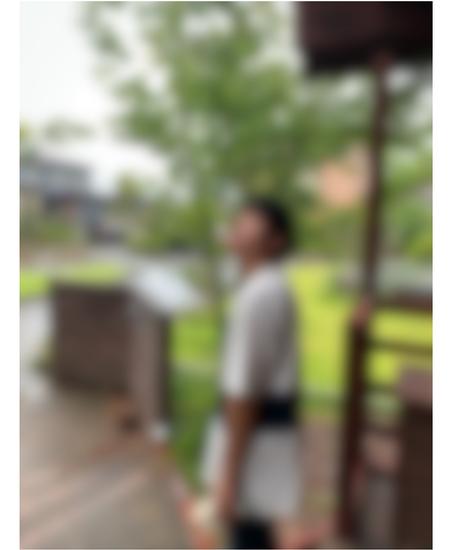
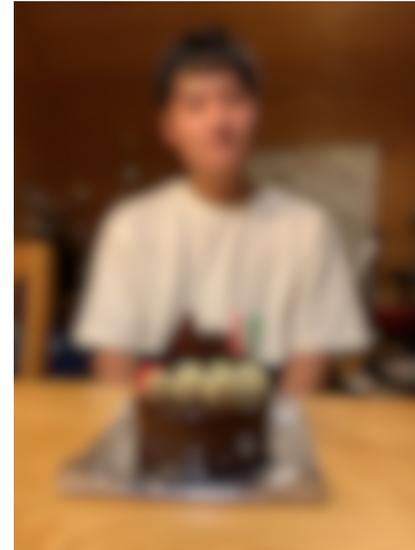
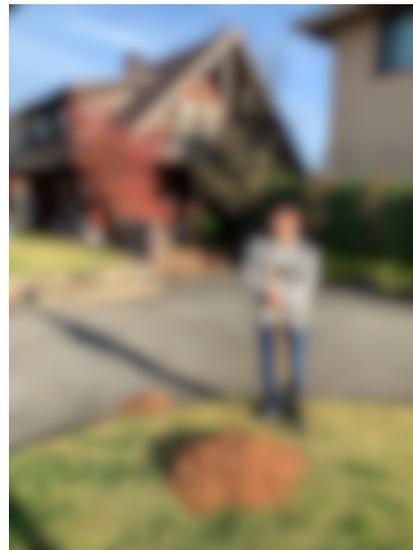
2004年生まれ 高校生



(手)



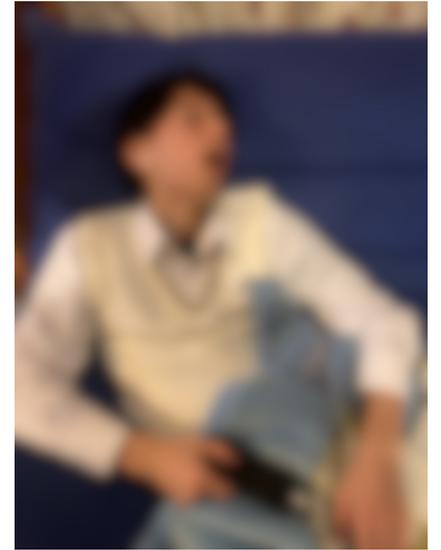
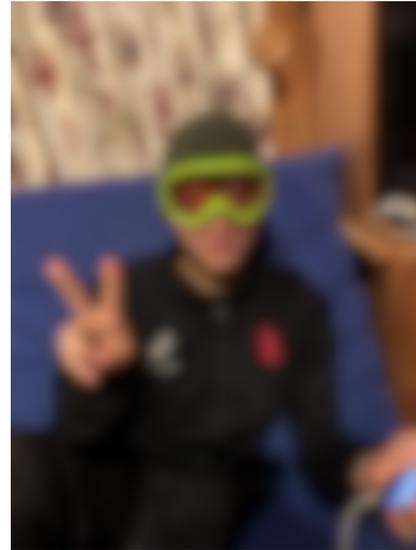
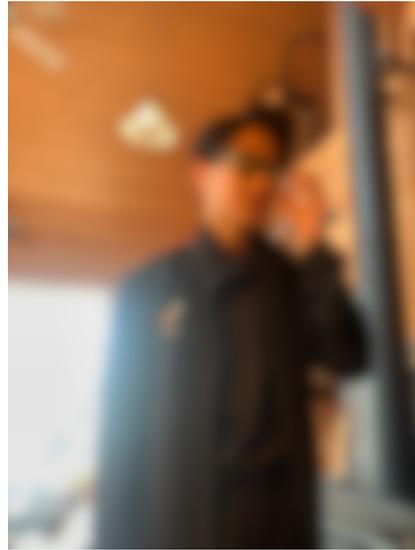
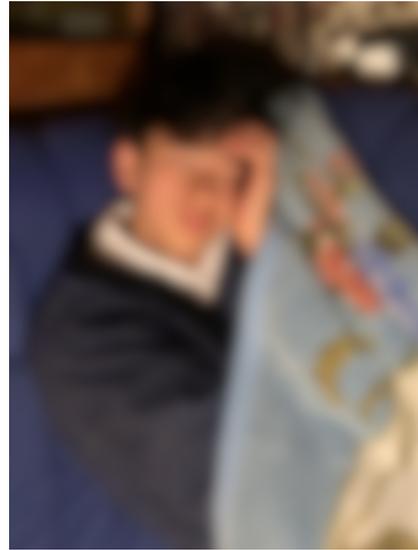
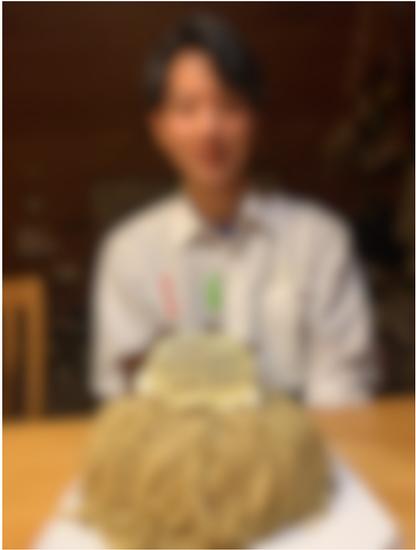
(帽子)



(半身)

# テストデータ：三男

2006年生まれ 高校生



(睡眠中)

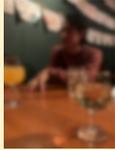
(サングラス)

(ゴーグル、帽子)

(睡眠中)

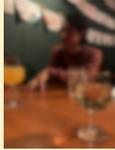
# 結果：長男

(数字 = 同一判定・類似度、x = 別人判定)

					
	99.997				
	99.999	99.996			
	99.991	99.914	99.968		
	99.978	99.512	99.805	99.995	

# 結果：長男

(数字 = 同一判定・類似度、x = 別人判定)

					
	99.997				
	99.999	99.996			!!
	99.991	99.914	99.968		
	99.978	99.512	99.805	99.995	

Perfect

# 結果：次男

(数字 = 同一判定・類似度、x = 別人判定)

					
	99.950				
	99.982	99.967			
	99.994	99.898	99.997		
	99.950	99.874	99.984	99.964	

# 結果：次男

(数字 = 同一判定・類似度、x = 別人判定)

					
	99.950				
	99.982	99.967			!!
	99.994	99.998	99.997		
	99.950	99.874	99.984	99.964	

Perfect

# 結果：三男

(数字 = 同一判定・類似度、x = 別人判定)

					
	99.886				
	99.979	99.318			
	99.533	97.448	99.542		
	99.878	99.855	99.757	91.801	

# 結果：三男

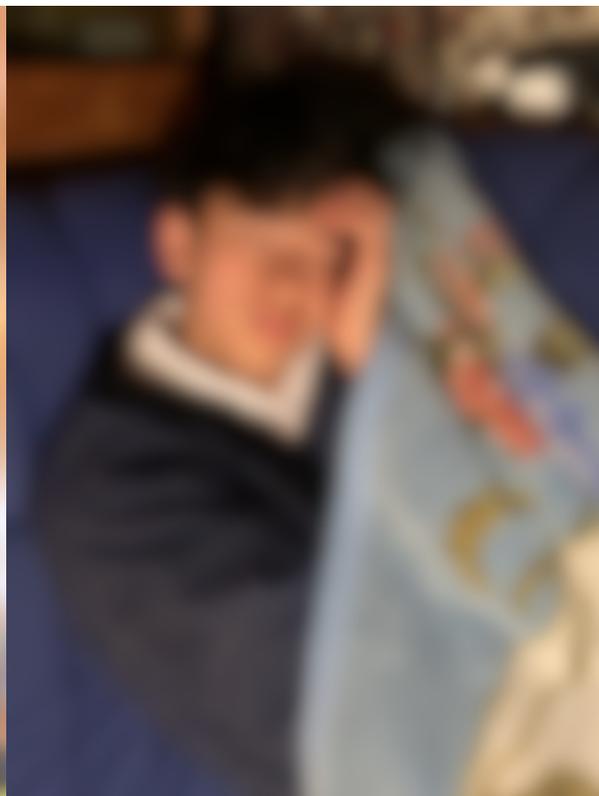
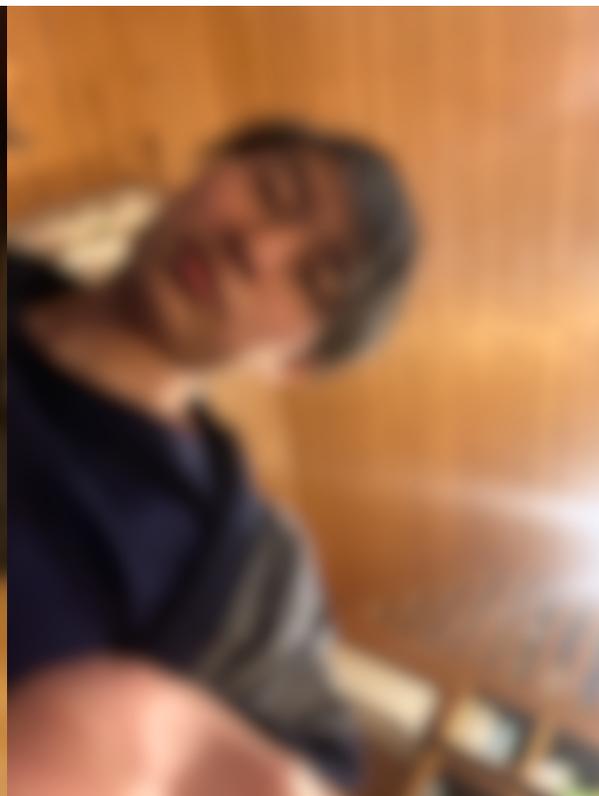
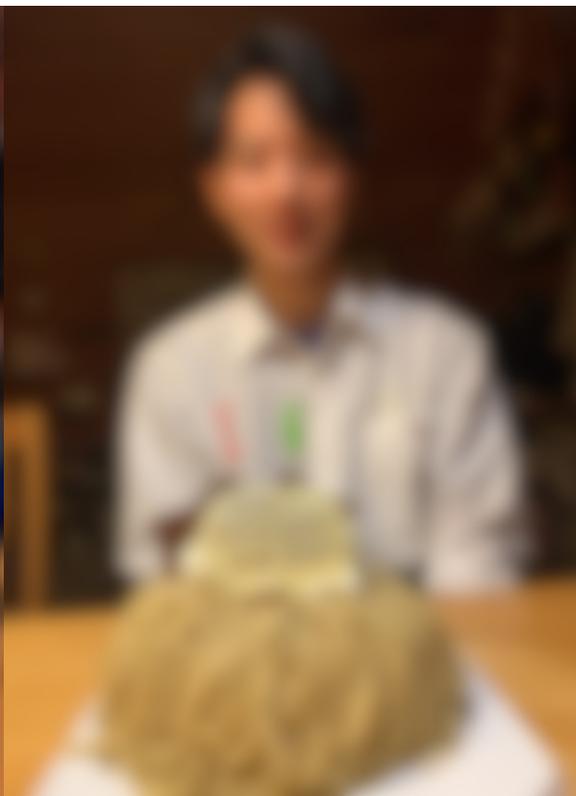
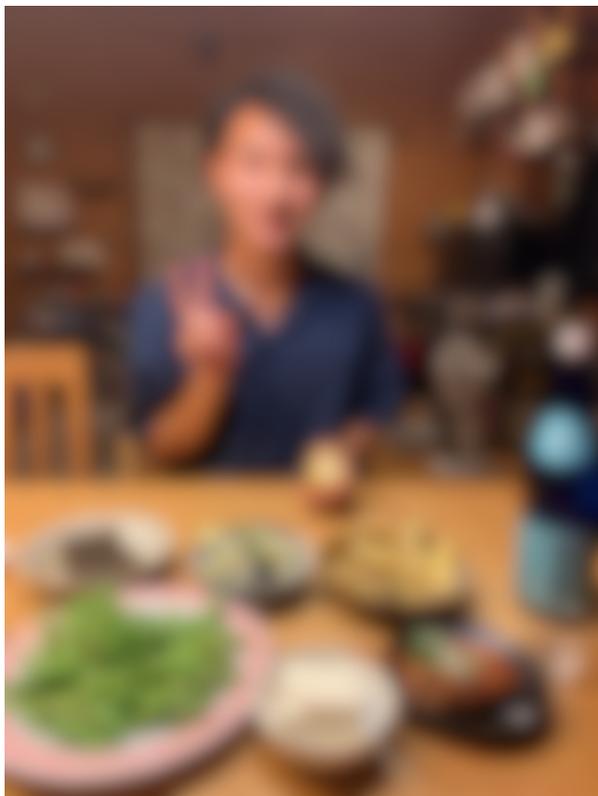
(数字 = 同一判定・類似度、x = 別人判定)

					
	99.886				
	99.979	99.318			!!
	99.533	97.148	99.542		
	99.878	99.855	99.757	91.801	

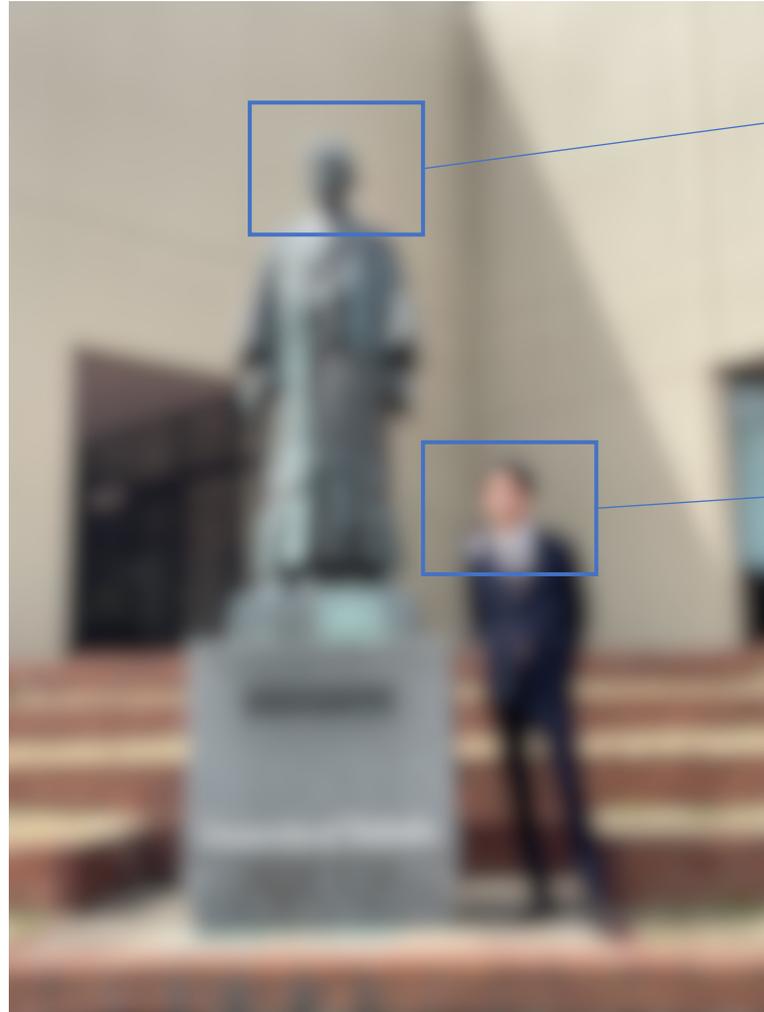
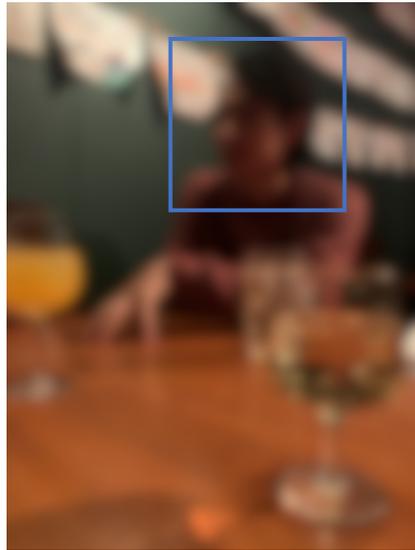




似てる・かな？



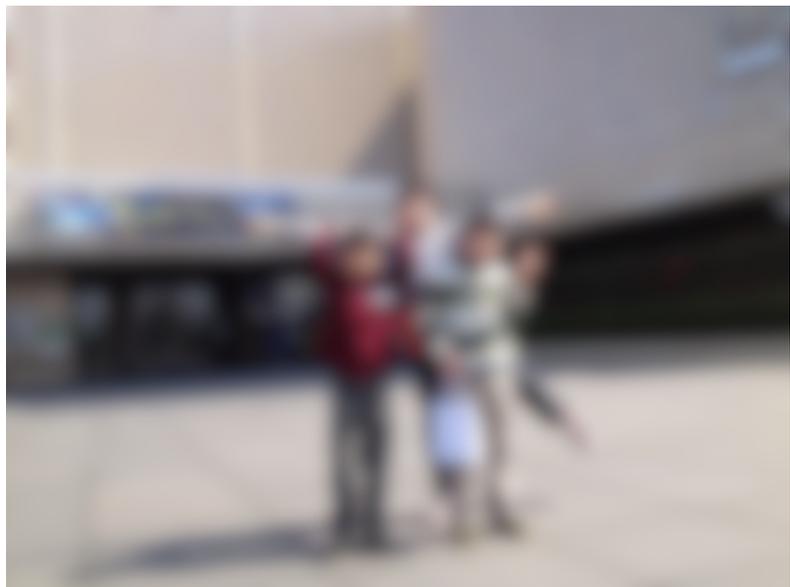
# 複数の顔もそれぞれ認識されます



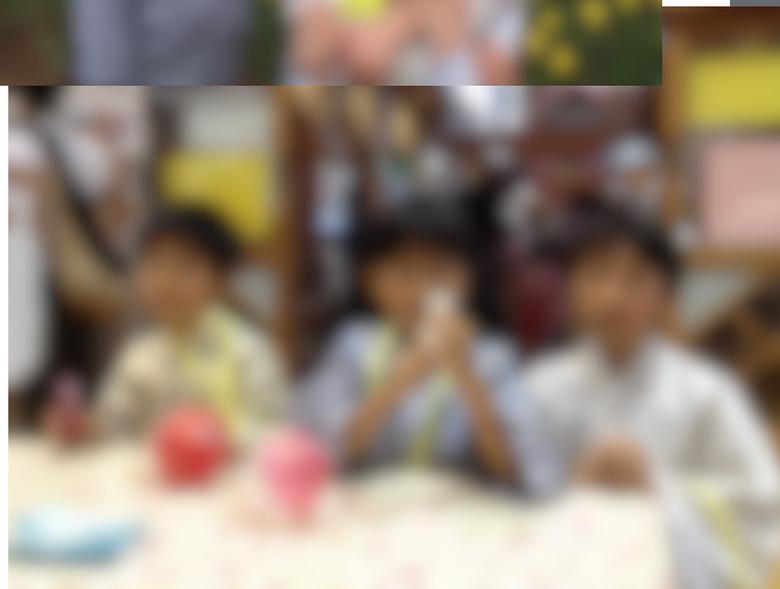
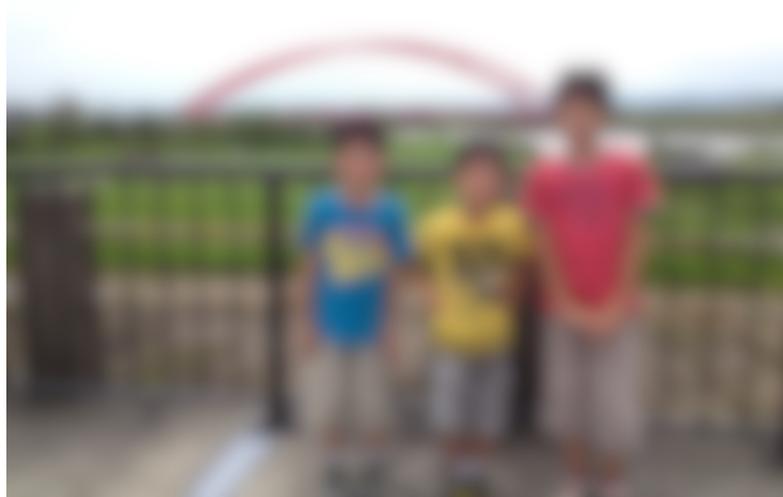
別人  
(嘉納治五郎)

同一人物

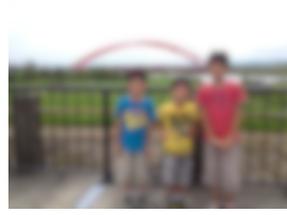
# 子供の頃の写真はどうかかな？



いずれも2013年頃の  
の写真



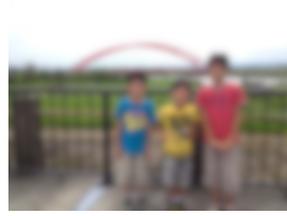
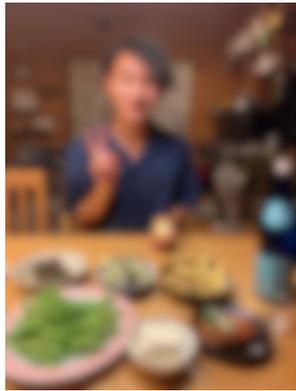
# 結果：長男



(数字 = 同一判定・類似度、x = 別人判定)

長男	99.414	96.967	98.176	98.750	94.368
次男	x	x	x	x	x
三男	x	x	x	x	x

# 結果：長男

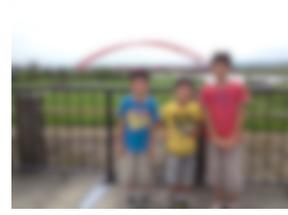
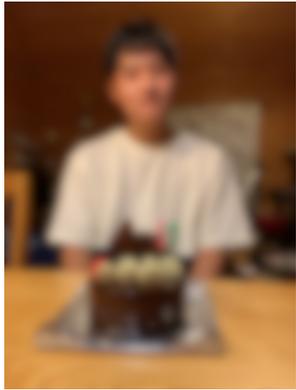


(数字 = 同一判定・類似度、x = 別人判定)

長男	99.414	96.967	98.176	98.750	94.368
次男	x	x	x	x	x
三男	x	x	x	x	x

Perfect!!!

# 結果：次男



(数字 = 同一判定・類似度、x = 別人判定)

長男	x	x	x	x	x
次男	99.938	98.222	98.602	99.885	98.325
三男	x	x	x	x	x

# 結果：次男

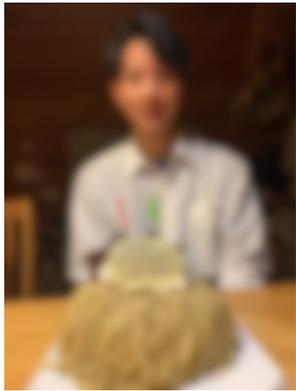


(数字 = 同一判定・類似度、x = 別人判定)

長男	x	x	x	x	x
次男	99.938	98.222	97.602	99.885	98.325
三男	x	x	x	x	x

Perfect!!!

# 結果：三男

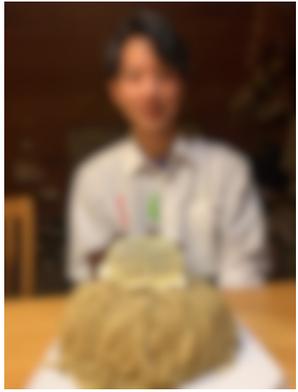


(数字 = 同一判定・類似度、x = 別人判定)

長男	82.262	x	x	x	x
次男	x	x	x	x	x
三男	93.420	84.656	81.902	99.091	95.387

# 結果：三男

やはり三男と長男は似ているのか・・・？

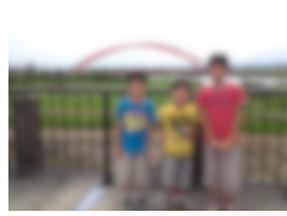
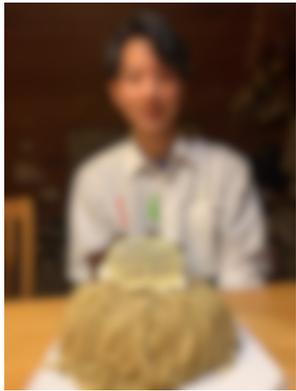


(数字 = 同一判定・類似度、x = 別人判定)

長男	82.262	x	x	x	x
次男	x	x	x	x	x
三男	93.420	84.656	81.902	99.091	95.387

結果：三男

類似度スコアが低い  
「変顔」の影響??



(数字 = 同一判定・類似度、x = 別人判定)

長男	82.262	x	x	x	x
次男	x	x	x	x	x
三男	93.420	84.656	81.902	99.091	95.387

# 誤検知を減らすには

類似度スコアに閾値を設けて、閾値に満たないものは一致と認めない、ということにすれば誤検知は減る（その副作用として「見逃し」が増える）

長男1	長男1																			
長男2	99.997	長男2																		
長男3	99.999	99.996	長男3																	
長男4	99.991	99.914	99.968	長男4																
長男5	99.978	99.512	99.805	99.995	長男5															
次男1	-	-	-	-	-	次男1														
次男2	-	-	-	-	-	99.950	次男2													
次男3	-	-	-	-	-	99.982	99.967	次男3												
次男4	-	-	-	-	-	99.994	99.898	99.997	次男4											
次男5	-	-	-	-	-	99.95	99.874	99.984	99.964	次男5										
三男1	92.341	-	-	83.212	-	-	-	-	-	-	三男1									
三男2	94.033	-	-	91.510	-	-	-	-	-	-	99.886	三男2								
三男3	-	-	-	-	-	-	-	-	-	-	99.979	99.318	三男3							
三男4	-	-	-	-	-	-	-	-	-	-	99.533	97.448	99.542	三男4						
三男5	-	-	-	-	-	-	-	-	-	-	99.878	99.855	99.757	91.801	三男5					

【例】 閾値を95%とした場合  
誤検知 = 0、見逃し = 5

	長男A	長男B	長男C	長男D	長男E	次男A	次男B	次男C	次男D	次男E	三男A	三男B	三男C	三男D	三男E
長男1	99.414	96.967	98.176	98.750	94.368	-	-	-	-	-	-	-	-	-	-
次男4	-	-	-	-	-	99.938	98.222	98.602	99.885	98.325	-	-	-	-	-
三男1	82.262	-	-	-	-	-	-	-	-	-	93.42	84.656	81.902	99.091	95.387

# 料金

リージョンによって微妙に値段が異なります  
今回利用したCompareFacesは「グループ1 API」です

リージョン: 米国東部 (オハイオ) ⇅

ボリューム (画像/月)	グループ 1 API で処理された画像	グループ 2 API で処理された画像
最初の 1,000,000 ページ	画像あたり 0.001USD	画像あたり 0.001USD
次の 4,000,000 ページ	画像あたり 0.0008USD	画像あたり 0.0008USD
次の 30,000,000 ページ	画像あたり 0.0006USD	画像あたり 0.0006USD
3,500,000 枚以上の画像	画像あたり 0.0004USD	画像あたり 0.00025USD

リージョン: アジアパシフィック (東京) ⇅

ボリューム (画像/月)	グループ 1 API で処理された画像	グループ 2 API で処理された画像
最初の 1,000,000 ページ	画像あたり 0.0013USD	画像あたり 0.0013USD
次の 4,000,000 ページ	画像あたり 0.001USD	画像あたり 0.001USD
次の 30,000,000 ページ	画像あたり 0.0008USD	画像あたり 0.0008USD
3,500,000 枚以上の画像	画像あたり 0.0005USD	画像あたり 0.0003125USD

# 料金

リージョンによって微妙に値段が異なります  
今回利用したCompareFacesは「グループ1 API」です

リージョン:	米国東部 (オハイオ) ⇅	
ボリューム (画像/月)	グループ 1 API で処理された画像	グループ 2 API で処理された画像
最初の 1,000,000 ページ	画像あたり 0.001USD	画像あたり 0.001USD
次の 4,000,000 ページ	画像あたり 0.0008USD	画像あたり 0.0008USD
次の 30,000,000 ページ	画像あたり 0.0006USD	画像あたり 0.0006USD
3,500,000 枚以上の画像	画像あたり 0.0004USD	画像あたり 0.00025USD

リージョン:	アジアパシフィック (東京) ⇅	
ボリューム (画像/月)	グループ 1 API で処理された画像	グループ 2 API で処理された画像
最初の 1,000,000 ページ	画像あたり 0.0013USD	画像あたり 0.0013USD
次の 4,000,000 ページ	画像あたり 0.001USD	画像あたり 0.001USD
次の 30,000,000 ページ	画像あたり 0.0008USD	画像あたり 0.0008USD
3,500,000 枚以上の画像	画像あたり 0.0005USD	画像あたり 0.0003125USD

**1画像処理あたり 0.12円 (US-EAST-1)**

**100万回/月の処理で 120,000円**  
**500万回/月の処理で 504,000円**  
**3,500万回/月の処理で 2,664,000円**

# まとめ

- Rekognitionはとってもお手軽に使える
- 顔認識精度はとても高く、10年前の顔でも正確に判定する
- 我が家の長男と三男は顔が似ていることが裏付けられた？
- 顔認識以外の機能もそのうち試してみたい

# Any questions?

確かな未来を、確かな力で。



グローバルネットコアは AWS セレクトティアサービスパートナーです。