

令和7年度 特別研究報告書

日焼けリスクを通知するシステムの検討

龍谷大学 先端理工学部 知能情報メディア課程

Y220223 伊藤 仁

指導教員 三好 力 教授

研究梗概

普段生活する中で紫外線による肌への影響（日焼け）を意識することがある。旅行に行った時や海水浴を楽しんでいる時に紫外線センサーを持ち歩いて UV インデックスをチェックし続けることは難しいことから UV インデックスを常にモニタリングすることができない。本研究では紫外線センサーではなく日常的に携帯されているスマートフォンを使って UV インデックスを推定し、その UV インデックス、時間、肌タイプによって日焼けリスクが推測される場合に利用者へ通知を送る手法を提案した。そこで、日々の UV インデックスやスマートフォンで撮影した空の写真の Exif データを分析して導き出された近似曲線を用いて日焼けリスクが推測されるタイミングで利用者へ通知をすることで、休憩を促し日焼けによるダメージを抑えるという提案したシステムの実現に必要な機能が動作するか実験して確認した。

本研究は、紫外線の推定にあたって外部機器を使用せず、手軽に誰でも紫外線対策を行える仕組みであり、日常生活における日焼け防止支援システムとしての実現に寄与するものである。

目次

第 1 章はじめに.....	1
第 2 章既存技術.....	2
2.1 黒玉式熱中症指数計.....	2
2.2 CliMate.....	2
2.3 コンタクト接点付 温度指示発信計.....	3
2.4 既存技術まとめ.....	3
2.5 UV インデックス	4
2.6 紫外線の性質.....	4
2.7 肌の性質.....	5
2.8 日焼けのタイミング.....	6
2.9 Exif データ	6
第 3 章提案手法.....	7
3.1 フローチャート.....	8
第 4 章実験.....	9
4.1 実験環境.....	9
4.2 実験方法.....	10
4.3 実験 1	11
4.3.1 実験結果.....	12
4.3.2 実験結果を踏まえて.....	13
4.4 実験 2	15
4.4.1 実験結果.....	15
4.5 実験 3	16
4.5.1 実験結果.....	16
第 5 章考察.....	18
第 6 章まとめに.....	19
謝辞	20
参考文献	21
付録	22

第1章 はじめに

海水浴やスポーツなどを行った後に起こる日焼けは、肌の老化や痛みを引き起こす厄介な問題である。日焼けは見た目の変化だけでなく、日焼けした肌の痛みや炎症を引き起こし、場合によっては日常生活に支障をきたすこともある。しかし、太陽から出る紫外線は人の目には見えなく、感じることもできないので強さや影響が分かりにくい。

[1]より紫外線の皮膚への影響は、太陽にあたっすぐにみられる急性傷害と、長年にわたってあたり続けて現れる慢性傷害に分けて考えることができる。

急性傷害とは、紫外線で皮膚に炎症が起こり、真っ赤で痛い日焼け（サンバーン）として現れるものである。日光にあたっ数時間後から赤くひりひりとした炎症が起こり、8時間から24時間でピークとなり、2、3日で消えて行くが、あたりすぎたときは水ぶくれになってしまう。

慢性傷害とは、長年の紫外線による影響が皮膚のシミやしわ、時には良性、悪性の腫瘍として現れるものである。お年寄りの顔や手の甲にみられるこれらの変化は、一般に加齢による老化と思われがちだが、実は紫外線による慢性傷害の結果であり、光老化は加齢による自然の老化とは異なり、適切な紫外線防御対策により防ぐことができるものである。

日焼け対策として長袖の服を着ることや日焼け止めを塗ることが挙げられるが、紫外線の強さを把握していることが必要である。その日焼け対策を適切なタイミングで行うために気象庁のサイトを常に確認することや、紫外線センサーを持ち歩くことは負担となる場合が多い。

本研究では、スマートフォンが記録する Exif データを利用してリアルタイムで UV インデックスを推定し、その紫外線の量と時間を見て日焼けするリスクがある場合に通知を行うスマートフォンのアプリを提案する。

第2章 既存技術

既存技術として以下のようなものが存在する。

2. 1 黒玉式熱中症指数計



図 2.1 黒玉式熱中症指数計

[5]より、黒玉式熱中症指数計(図 2.1)は熱中症予防指針にあわせて「危険」「厳重警戒」「警戒」「注意」の4区分で熱中症の危険度を分類し、危険度に合わせて異なる警告アラームで通知する。

2. 2 CliMate

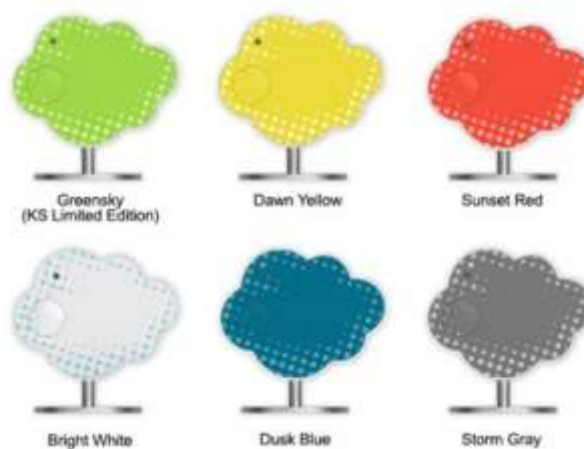


図 2.2 CliMate

[6]より、CliMate(図 2.2)は湿度・紫外線・気温測定をしてくれる小型ガジェットで、フックをつければバッグなどに取り付けて持ち運びも可能である。測定結果はBluetoothを通じてアプリで確認でき、水分補給や日焼け止めクリームを塗るタイミングまで教えてくれる。

2. 3 コンタクト接点付 温度指示発信計



図 2.3 コンタクト接点付 温度指示発信計

[7]より、コンタクト接点付 温度指示発信計(図 2.3)は現場型ダイヤル指示温度計に、接点および温度センサを組み込んだ製品である。コンタクト接点付き温度計は、精度の高い指示計にコンタクト機構を内蔵したもので、微細な温度変化にも敏感に反応し、ベル・ランプ等による警報・監視で作業の安全確保、機械装置の能率向上、作業標準による品質管理を行う事が出来る。

2. 4 既存技術まとめ

このように気温を測って警告、監視するものや紫外線を測定する既存技術はあるが、スマートフォンだけでリスクを検知するものは存在していない。

2. 5 UV インデックス

[2]より UV インデックス(図 2.4)とは紫外線が人体に及ぼす影響の度合いをわかりやすく示すために、紫外線の強さを指標化したもので、紫外線はUV-C、UV-B、UV-A の三つに分かれている。

UV-C は大気層(オゾンなど)で吸収され、地表には到達しない。UV-B のほとんどは大気層(オゾンなど)で吸収されるが、一部は地表へ到達し、皮膚や眼に有害である。日焼けを起こしたり、皮膚がんの原因となる。UV-A はUV-B ほど有害ではないが、長時間浴びた場合の健康影響が懸念されている。

そしてこの UV-A などは波長で分かれており、波長が 100 ナノメートルから 280 ナノメートルを UV-C、280 ナノメートルから 315 ナノメートルを UV-B、315 ナノメートルから 400 ナノメートルを UV-A としている。

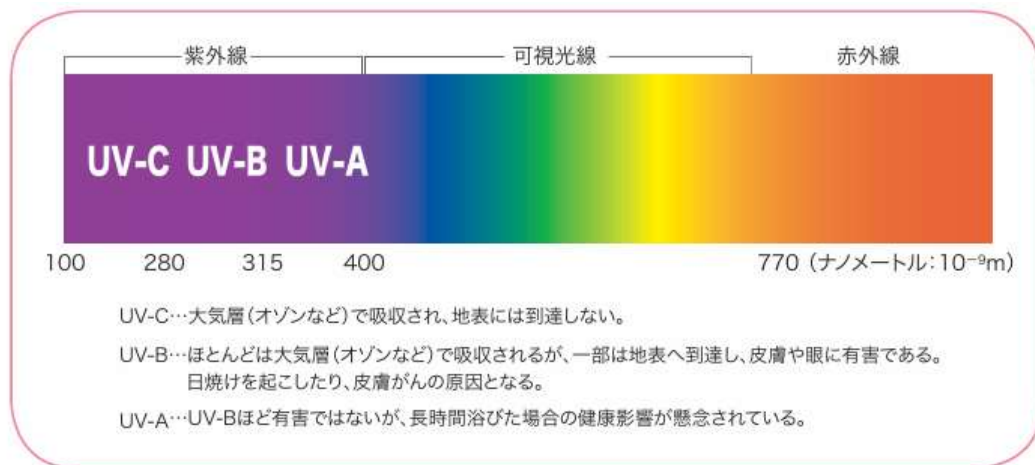


図 2.4 UV インデックス

2. 6 紫外線の性質

[1]より紫外線の性質(図 2.5)は雲や標高などによって変化する。

①薄い雲では UV-B の 80~90%が透過し、屋外では太陽から直接届く紫外線量と空気中で散乱して届く紫外線量がほぼ同程度になっている。

②地表面の種類により紫外線の反射率は大きく異なる(新雪 80%、砂浜 10~20%、コンクリート・アスファルト 10%、水面 10~20%、草地・芝生、土面 10%以下)。

③標高が 1000m 上昇するごとに紫外線量は 10~12%増加する。

④南に行く(緯度が低くなる)ほど強い。

⑤1年のうちでは春から初秋にかけて強い。4~9月に1年間のおよそ 70~80%

⑥1日のうちでは正午をはさむ数時間が強い。夏の午前 10 時~午後 2 時に 1 日のおよそ 70%冬の同じ時間帯では 1 日の照射量の 80~85%を占める。

紫外線の性質	
①	薄い雲ではUV-Bの80～90%が透過します。屋外では太陽から直接届く紫外線量と空気中で散乱して届く紫外線量がほぼ同程度になっています。
②	地表面の種類により紫外線の反射率は大きく異なります(新雪80%、砂浜10～20%、コンクリート・アスファルト10%、水面10～20%、草地・芝生、土面10%以下)。
③	標高が1000m上昇するごとに紫外線量は10～12%増加します。
④	年間で見ると、屋内で働く人は屋外で働く人の10～20%の紫外線を浴びています。

(WHO : Protection against exposure to ultraviolet radiation 1995)

図 2.5 紫外線の性質

2. 7 肌の性質

[3]より肌の性質（図 2.6）は1から6まであり、日本人は2から4の間の肌タイプの人が多いと言われている。そして、本研究では人々の肌タイプ3（UV インデックス3から5で30分紫外線を浴びると日焼けしていく）として考えて実験していこうと思う。

SKIN TYPE CLASSIFICATION		BURNS IN THE SUN	TANS AFTER HAVING BEEN IN THE SUN
I.	Melano-compromised	Always	Seldom
II.		Usually	Sometimes
III.	Melano-competent	Sometimes	Usually
IV.		Seldom	Always
V.	Melano-protected	Naturally brown skin Naturally black skin	
VI.			

図 2.6 肌の性質

2. 8 日焼けのタイミング

[8]より、図 2.7 では UV インデックスが 6 から 8 程度で日焼けするまでの時間である。

時間帯によって UV インデックスが上下するため正確な時間を求めることは難しいが、日本人の肌は肌タイプⅡからⅣの人が多いため 30 分前後で日焼けしてしまうと考えられる。

Skin type	Maximum amount of time
I	10 minutes
II	20 minutes
III	30 minutes
IV	50 minutes
V	more than 60 minutes
VI	more than 60 minutes

図 2.7 日焼けのタイミング

2. 9 Exif データ

[4]より Exif データとは写真の中で画像以外の様々なデータが記録されたものである。

Exif データの中には撮影日時、カメラのメーカーや機種、絞り値、シャッタースピード、ISO 感度、焦点距離、GPS 情報などの様々な情報が入っている。

第3章 提案手法

本研究では、利用者が日常的に所持しているスマートフォンのみを用いて、日焼けリスクを簡単に把握し、日焼けリスクを防止する行動を促すことを目的とする。利用者は特別な操作や外部機器を必要とせず、スマートフォンを机の上などに上向きに設置するだけで利用可能である。スマートフォンが記録する Exif データを利用して UV インデックスを推定し、その紫外線の量と時間を見て日焼けするリスクがある場合に通知を行うアプリを作成する。

まず、スマートフォンと紫外線センサーを用意して空の写真とその方向からの紫外線の UV インデックスを記録する。そして、空の写真に入っている Exif データと紫外線センサーで計測した UV インデックスとの関係を調べる。その結果を用いて紫外線の量と時間を見て日焼けするリスクがある場合に通知を行うことで日焼けによる被害を抑制できると考えられる。紫外線センサーを使用し、空の写真を撮ると同時に UV インデックスを計測する。その際、紫外線センサーの先とスマホの撮影角度を合わせることが重要である。

空の写真の Exif データから、絞り値、ISO 感度、シャッタースピードの三つの値を UV インデックスと同時に記録していく。上記で記録したデータをもとに近似曲線を作成し、その結果を用いて利用者の日焼けリスクを推定する。

推定の結果、リスクが高い場合は、アプリを通じて日焼けリスクを通知する。

工程のフローチャートを図 3, 1 に示す。

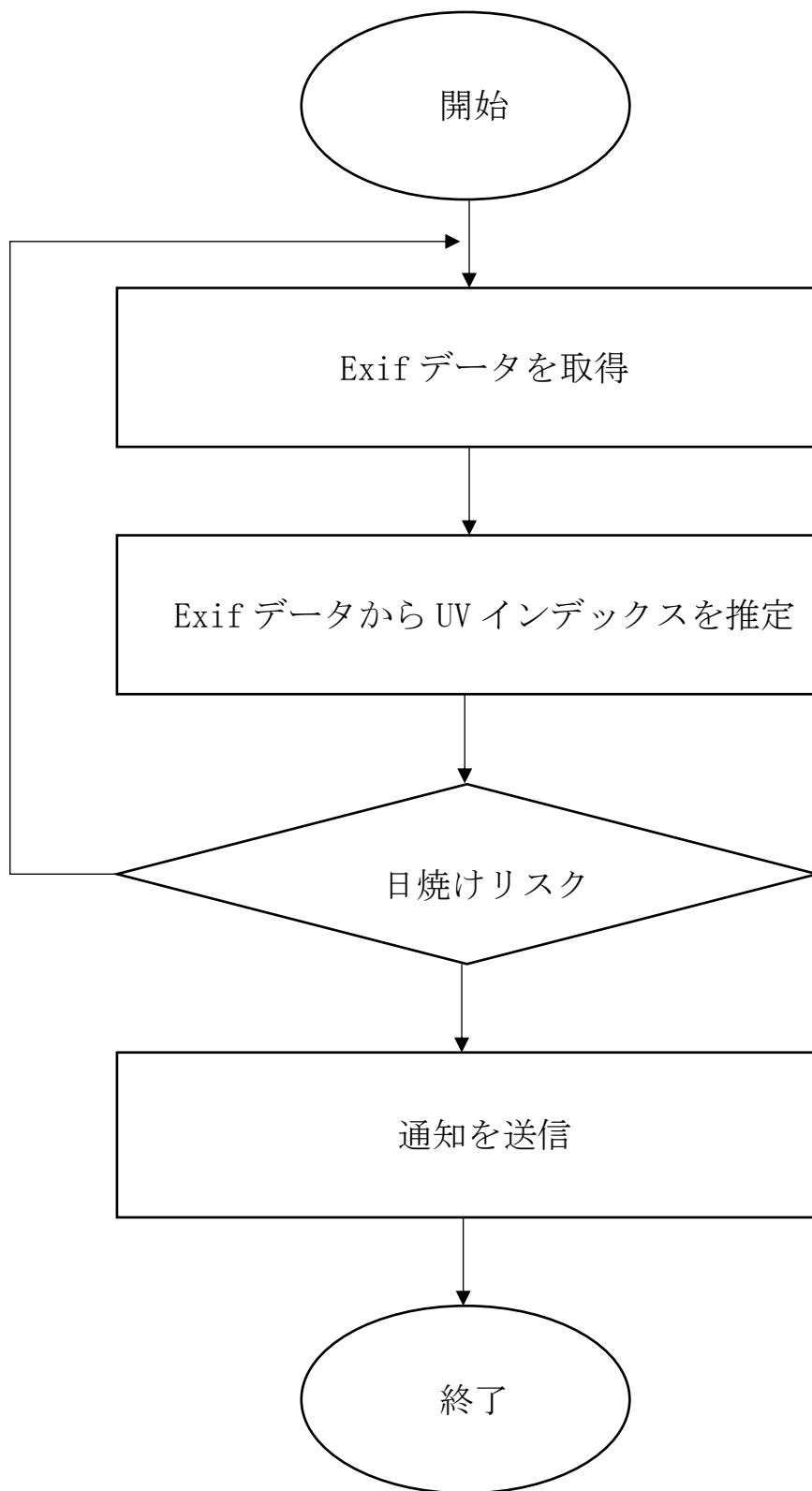


図 3.1 システムのフローチャート

第4章 実験

本実験では、スマートフォンのみでUVインデックスを推定することが目的である。なので、Exifデータと紫外線センサーによるデータを分析し、その後にスマホのみでUVインデックスを推定する実験を行った。

4. 1 実験環境

本実験に使用したスマートフォンは
Android Pixel 7 Pro



紫外線センサーは
BQBQERT KF-90



使用したスマートフォンは iPhone 16e



4. 2 実験方法

スマートフォンを屋外の直射日光が当たる場所に、画面を上向きにして机の上に設置する。その状態で写真を撮影し、撮影時刻、ISO 感度、露出時間、絞り値などの Exif データを取得する。

私の場合はこのようにして空を撮影している。



4. 3 実験1

晴れている日、薄曇りの日、曇りの日を選んで自宅のベランダで撮影した。空の方向の写真撮影し、その後同じ向きに紫外線センサーで測定した。測定結果を Excel に記入した。



4. 3. 1 実験結果

4, 3 の工程で測定した Exif データを分析した結果を表 4. 3. 1. 1 に示す。

表 4. 3. 1. 1 実験結果

天気	ISO	F値	シャッター速度	UV	
薄曇り	48	1.9	0.000114995	9	
晴れ	48	1.9	0.000114995	11	
晴れ	47	1.9	0.000114995	10	
晴れ	38	1.9	0.00008	15	
晴れ	53	1.9	3.50005E-05	15	
曇り	45	1.9	0.000512033	4	
曇り	37	1.9	0.00058309	3	
曇り	45	1.9	0.001069519	0	
曇り	45	1.9	0.000114995	5	
薄曇り	40	1.9	0.000149993	7	
薄曇り	42	1.9	9.70026E-05	9	
薄曇り	47	1.9	0.000317965	7	
曇り	47	1.9	0.000362056	5	
薄曇り	40	1.9	0.000344947	6	
曇り	47	1.9	0.000477099	4	
薄曇り	38	1.9	0.00030003	6	
薄曇り	42	1.9	0.000167729	8	
薄曇り	45	1.9	0.000176991	8	
曇り	40	1.9	0.000750751	4	

4. 3. 2 実験結果を踏まえて

測定した Exif データでは、ISO 感度と F 値の数値がほとんど変わらず、写真の明るさを示しているのはシャッター速度だと考えたため UV インデックスを推定するのにシャッター速度のみを用いたものを表 4. 3. 2. 1 と図 4. 3. 2. 2 に示す。

表 4. 3. 2. 1 実験結果

天気	シャッター速度	UV
薄曇り	0.000114995	9
晴れ	0.000114995	11
晴れ	0.000114995	10
晴れ	0.00008	15
晴れ	3.50005E-05	15
曇り	0.000512033	4
曇り	0.00058309	3
曇り	0.001069519	0
曇り	0.000114995	5
薄曇り	0.000149993	7
薄曇り	9.70026E-05	9
薄曇り	0.000317965	7
曇り	0.000362056	5
薄曇り	0.000344947	6
曇り	0.000477099	4
薄曇り	0.00030003	6
薄曇り	0.000167729	8
薄曇り	0.000176991	8
曇り	0.000750751	4

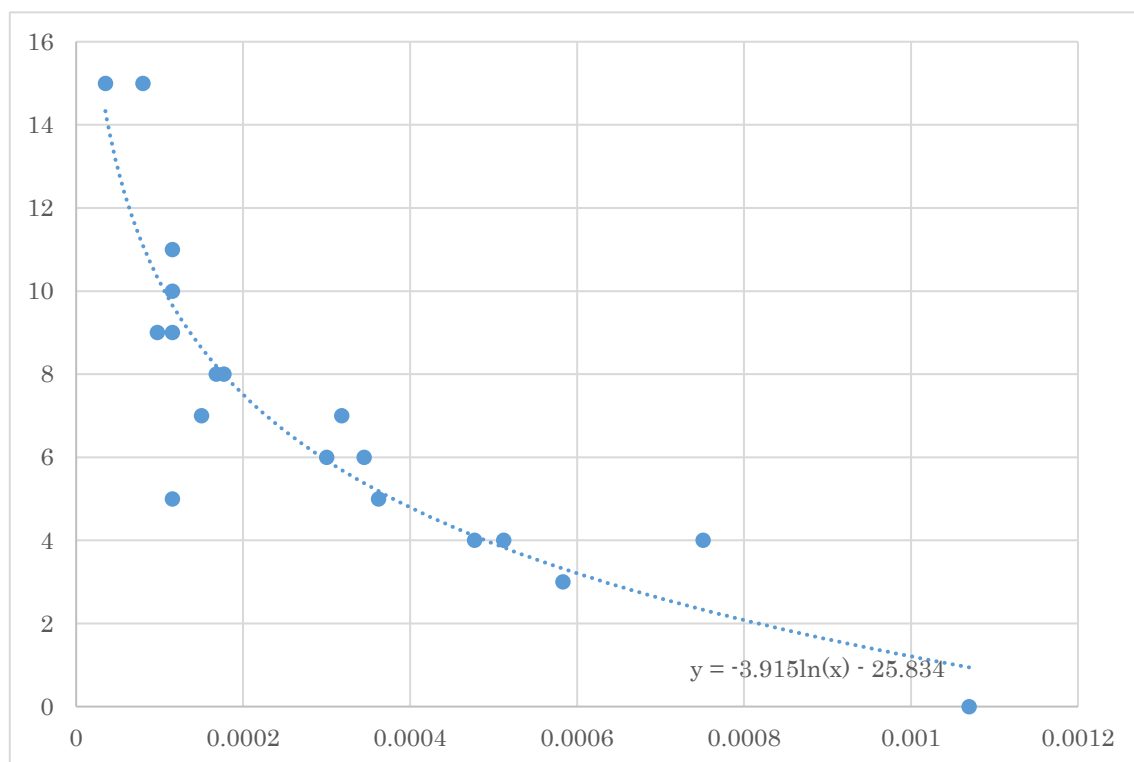


圖 4. 3. 2. 2 分析結果

4. 4 実験2

夏だけでなく秋でも UV インデックスを推定するために実験する。スマートフォンは Android Pixel 7 Pro で第 4 章の実験と同じ方法で UV インデックスと Exif データを分析して近似曲線を求めたものを表 4. 4. 1. 1 と図 4. 4. 1. 2 に示す。

4. 4. 1 実験結果

表 4. 4. 1. 1 実験結果

天気	シャッター速度	UV
曇り	0.000264971	5
曇り	0.000336022	4
曇り	0.000521105	2
曇り	0.001254705	1
薄曇り	3.50005E-05	8
晴れ	0.00008	7
晴れ	3.50005E-05	9
晴れ	6.20001E-05	8
晴れ	0.000750751	3
晴れ	0.001626016	2
晴れ	0.003891051	0
晴れ	0.000406009	3
晴れ	0.000414938	4
晴れ	0.000433088	3

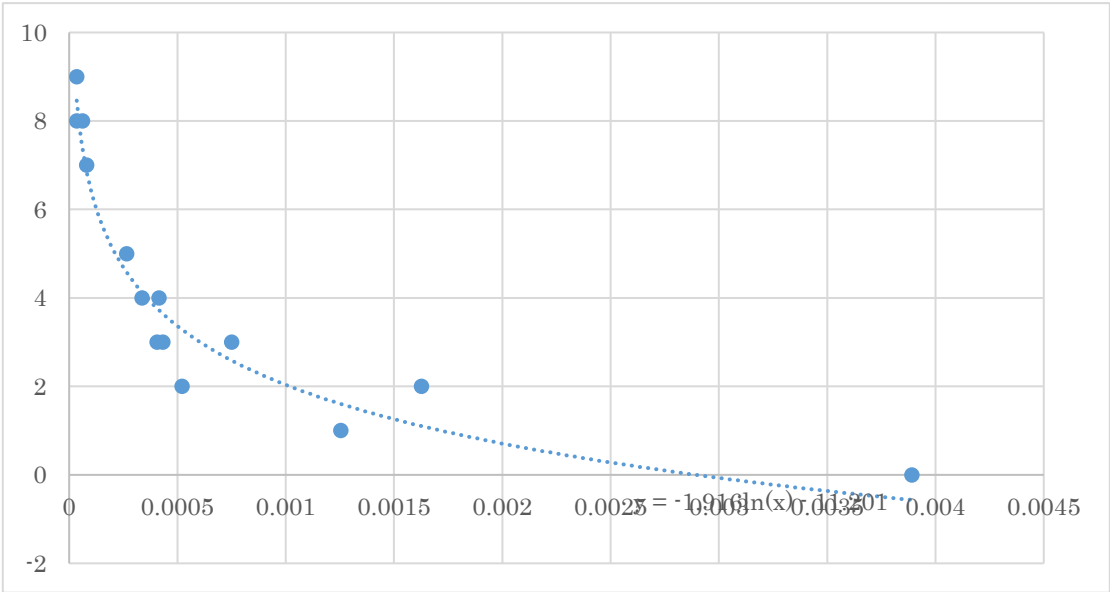


図 4. 4. 1. 2 分析結果

4. 5 実験3

実験1の結果を他のスマートフォンでも適用することで可能か確認するために実験2を行った。私が普段使用しているスマートフォンである iPhone 16e で第三章の実験1と同じ方法でUVインデックスとExifデータを分析して近似曲線を求めたものを表4.5.1.1と図4.5.1.2と図4.5.1.3に示す。

以下は4.4の実験2と同様に秋での実験結果である。

4. 5. 1 実験結果

表 4. 5. 1. 1 実験結果

シャッター速度	シャッター速度 (iPhone)	UV
0.001422475	0.002247191	1
0.001190476	0.001788909	2
0.000660939	0.000762195	2
0.001633987	0.001851852	0
0.00094518	0.000952381	3
0.000344947	0.000541126	3
0.000264971	0.000351989	4
0.000477099	0.001251564	3
0.000256016	0.000494071	4
0.000211999	0.000368053	5
0.00030003	0.000526039	4
0.000220994	0.000241022	4
0.000220994	0.000209996	5
0.000168011	0.000257003	6
0.000188005	0.000161996	5
0.00030003	0.000289017	4
0.000176991	0.000161996	6
0.000149993	0.000130993	7
0.000149993	0.000161996	6
0.000159008	0.000146994	7
0.000149993	0.000130993	7
0.000186012	0.000177999	6
0.000211999	0.000209996	5
0.000282965	0.000257003	4
0.000256016	0.000304044	4
0.000211999	0.000241022	5
0.001254705	0.001172333	2
0.002155172	0.001631321	0
0.00148368	0.001347709	1
0.002040816	0.002024291	1

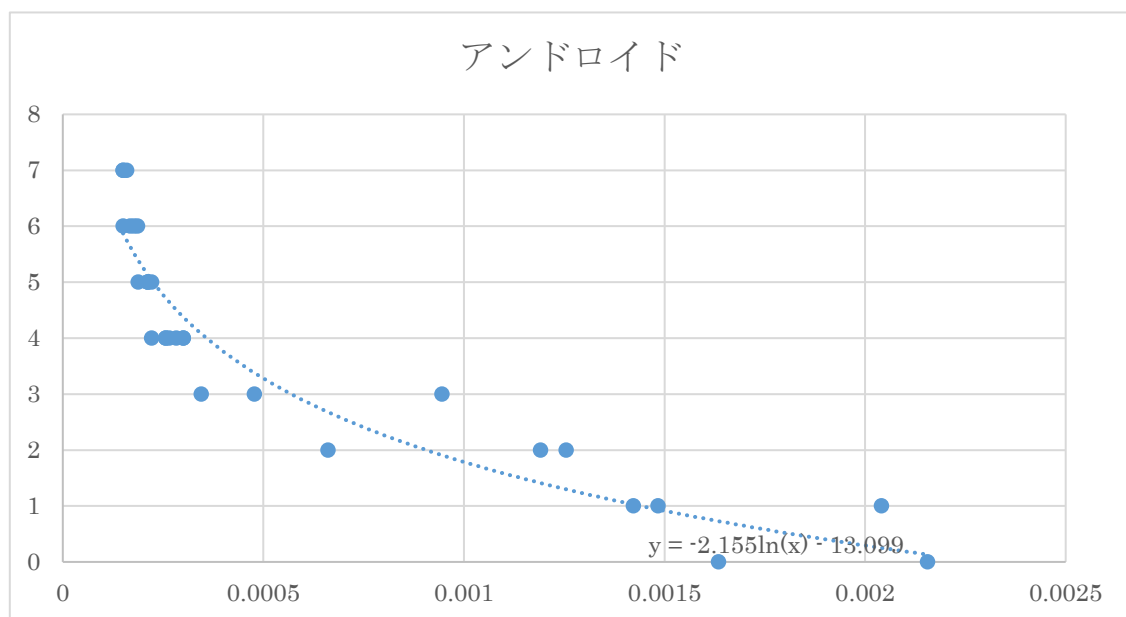


図 4.5.1.2 分析結果

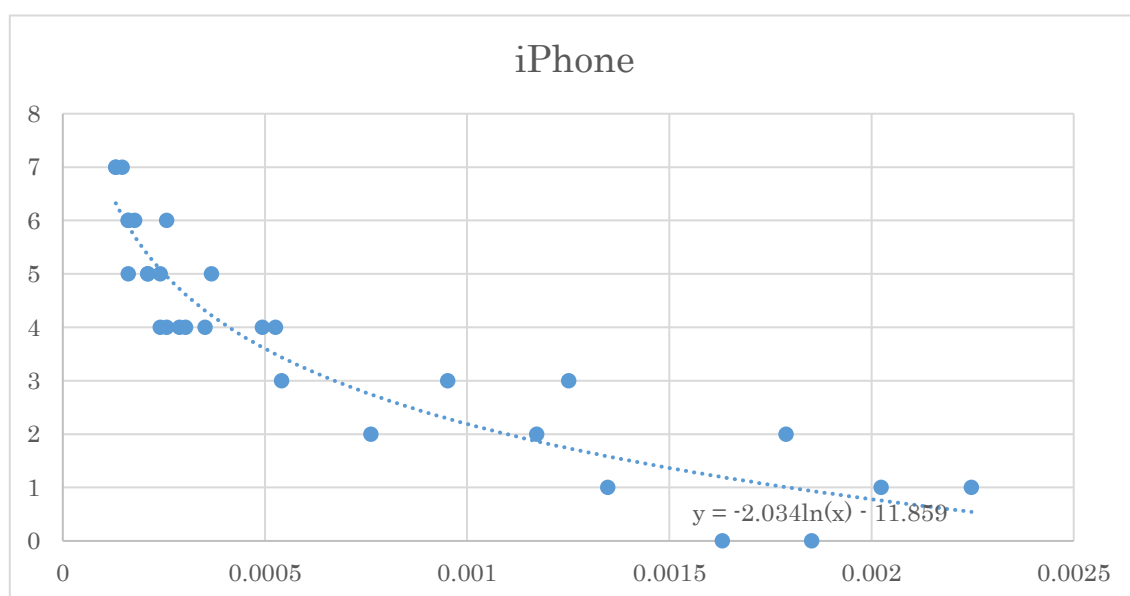


図 4.5.1.3 分析結果

第5章 考察

本研究では Exif データのシャッター速度から UV インデックスを推定することが可能であることが分かった。第4章の実験2より、実験の期間が延びてしまうと UV インデックスの変化してしまうことがあることが分かった。このようなことになる原因として考えられるのは、地球の公転と地軸の傾きによって夏から秋にかけて太陽高度が低くなっていくことで UV インデックスが変わってしまうことが原因だと考えられる。

第4章の実験3より、iPhone とアンドロイドで同じようなグラフになったのだが、シャッター速度が iPhone の方が長くなっている。その理由は、iPhone の F 値が 1.64 でアンドロイドの F 値が 1.9 だから（F 値が大きいほどレンズが閉じているのでシャッター速度が長い）という理由が考えられる。

このような実験の結果から季節ごとに異なる近似式を用いることやスマートフォンのメーカーや機種ごとに正しい UV インデックスが推測できるように調整する仕組みを作り上げることが必要である。

第6章 まとめに

本研究では、スマートフォンに記録される Exif データを利用して UV インデックスを推定し、日焼けのリスクが高い場合に利用者へ通知を行うシステムを提案した。

実験では、スマートフォンで空の写真を撮影すると同時に紫外線センサーで UV インデックスを測定することで、Exif データとインデックスの関係を分析した。その結果、ISO 感度や絞り値は大きな変化が見られなかったが、シャッター速度は UV インデックスの変化と強い関係があることが分かった。このシャッター速度を用いた近似式により、スマートフォンだけで UV インデックスを推定できる可能性が示された。

また、夏と秋で異なる近似式が必要となることや、スマートフォンの機種によってシャッター速度の値が異なることから、季節やデバイスを考慮する必要性も明らかになった。今後は、より多くのデータを長期間収集し、推定精度の向上を図るとともに、通知機能を実装して実際の日焼け防止効果を検証することが課題である。

謝辞

今回の研究、論文作成を行うにあたり、ご指導いただきました三好力教授に深謝いたします。また、多忙の中、日頃の議論にご協力くださった同三好研究室の皆様や、学友の皆様に心より感謝いたします。

参考文献

- [1]環境省”紫外線環境保健マニュアル”(900410650.pdf) (2020)
- [2]気象庁(<https://www.jma.go.jp/jma/kishou/known/env/ozonehp/3-0ozone.html>)
- [3]WHO ([Global solar UV index. A Practical Guide](#)) (2002)
- [4]Adobe ([EXIF ファイルの詳細と開く方法 | Adobe](#)) (2025)
- [5]TANITA, ” TT-562/562N”([黒球式熱中症指数計 熱中アラーム TT-562/562N | タニタ](#)) (2023)
- [6]Kickstarter, PBC © 2025, ” CliMate- Create your own friendly environment”
[CliMate- Create your own friendly environment by Rooti — Kickstarter](#), (2015)
- [7] 株式会社荏原計器製作所 ([製品案内-No.2-リーフレット.indd](#)) (2025)
- [8]IQWiG ([In brief: How much sun is too much? – InformedHealth.org – NCBI Bookshelf](#)) (2025)

付録

```
package com.example.uvindexapp2

import android.Manifest
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Bundle
import android.provider.MediaStore
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.exifinterface.media.ExifInterface
import java.io.InputStream
import kotlin.math.log2

class MainActivity : AppCompatActivity() {

    private lateinit var resultView: TextView
    private val PERMISSION_REQUEST = 100
    private val PICK_IMAGE = 101

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        resultView = TextView(this).apply {
            textSize = 18f
            setPadding(20, 50, 20, 20)
        }
        setContentView(resultView)

        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_MEDIA_IMAGES)
            != PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(
                this,

                arrayOf(Manifest.permission.READ_MEDIA_IMAGES),
                    PERMISSION_REQUEST
                )
        } else {
            pickImage()
        }

        override fun onRequestPermissionsResult(
            requestCode: Int, permissions: Array<out String>,
            grantResults: IntArray
        ) {
            super.onRequestPermissionsResult(requestCode,

permissions, grantResults)
                if (requestCode == PERMISSION_REQUEST) {
                    if (grantResults.isNotEmpty()) &&
grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                        pickImage()
                    } else {
                        Toast.makeText(this, "画像読み込み権限が
ありません", Toast.LENGTH_LONG).show()
                    }
                }

                private fun pickImage() {
                    val intent = Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI)
                    startActivityForResult(intent, PICK_IMAGE)
                }

                override fun onActivityResult(requestCode: Int,
resultCode: Int, data: Intent?) {
                    super.onActivityResult(requestCode, resultCode,
data)

                    if (requestCode == PICK_IMAGE && resultCode ==
Activity.RESULT_OK && data != null) {
                        try {
                            val uri: Uri? = data.data
                            val inputStream: InputStream? =
contentResolver.openInputStream(uri!!)
                            val exif = ExifInterface(inputStream!!)

                            val fNumber =
exif.getAttributeDouble(ExifInterface.TAG_F_NUMBER, 2.0)
                            val exposureTime =
exif.getAttributeDouble(ExifInterface.TAG_EXPOSURE_TIME,
1/100.0)
                            val iso =
exif.getAttributeDouble(ExifInterface.TAG_ISO_SPEED_RATINGS,
100.0)

                            resultView.text = ""
                            F 値: $fNumber
                            シャッター速度: $exposureTime
                            ISO: $iso
                        } catch (e: Exception) {
                            Toast.makeText(this, " エ ラ ー :
${e.message}", Toast.LENGTH_LONG).show()
                        }
                    }
                }
            }
        }
    }
}
```