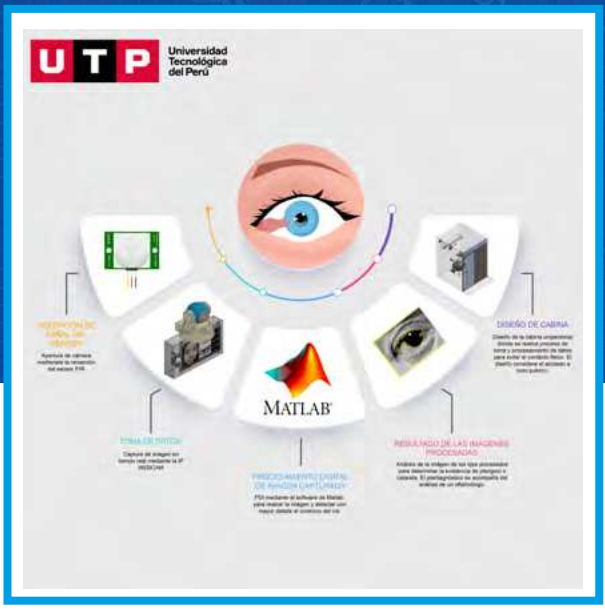
Implementación de procesamiento de imágenes en matlab para el prediagnóstico en oftalmología



Acosta, Williams; Puentes, Sharon; Temple, Lucero

Universidad Tecnológica del Perú, Lima – Perú

Contacto: willyacosta777@gmail.com

RESUMEN

La presente investigación busca explicar la implementación del procesamiento de imágenes en Matlab y su relación con el prediagnóstico de problemas oculares como base para automatizar el proceso de análisis de imágenes médicas en este campo. La pandemia ha generado una crisis de salud mundial y en nuestro país se ha observado un impacto significativo en el resultado de las teleconsultas, pues el desafío radica en la poca claridad al visualizar el ojo del paciente mediante una pantalla. Por tanto, se plantea la aplicación de una cabina unipersonal que permitirá la captura en tiempo real del ojo del paciente y su respectivo análisis. Además, para evitar el contacto, se implementa un sensor infrarrojo que detecta el ingreso de personas a la cabina. Finalmente, las técnicas de procesamiento brindan un panorama claro y demuestran que el proyecto puede adaptarse para situaciones específicas con los conocimientos de ingeniería mecatrónica y biomédica.

ABSTRACT

This research seeks to explain the implementation of image processing in Matlab and its relationship with the prediagnosis of eye problems as a basis for automating the process of medical image analysis in this field. The pandemic has generated a worldwide health crisis, and in our country a significant impact on the outcome of teleconsultations has been observed, since the challenge lies in the lack of clarity when visualizing the patient's eye through a screen. Therefore, we propose the application of a single-person booth that will allow real-time capture of the patient's eye and its respective analysis. In addition, to avoid contact, an infrared sensor is implemented to detect the entry of people into the booth. Finally, the processing techniques give a clear picture and demonstrate that the project can be adapted to specific situations with the knowledge of mechatronic and biomedical engineering.

Palabras clave: Matlab; procesamiento de imágenes; prediagnóstico; sensor.

INTRODUCCIÓN

El procesamiento de imágenes (PDI) constituye una representación tecnológica de alto impacto debido a los beneficios y resultados que propone en la actualidad. La implementación de esta herramienta agregada a la arquitectura de un sistema electrónico propicia el diagnóstico oportuno y una mejor visualización de imágenes médicas en oftalmología. Es cierto que Matlab fue incorporado en 1984; no obstante, en este contexto pandémico conviene conocer y explorar todos los recursos que pone a disposición de los futuros especialistas en ingeniería. De esa forma, Deperlıoğlu y Köse [1], demuestran que la función de convolución en Matlab puede tomar gran importancia en la detección y clasificación de 5 distintos tipos de enfermedades oculares. Además, representa una solución al tener la capacidad de identificar claramente las infecciones en la zona de la vista. Frente a esto, se puede señalar que el software Matlab integra diversas funcionalidades que, con ayuda de conocimientos empíricos revisados en ingeniería, se pueden adaptar a un contexto planteado. En este caso, existe una gran curiosidad por mejorar y automatizar los prediagnósticos que deben ser verificados por un médico especialista. Además, se toma en cuenta las grandes ventajas que supone tanto para los médicos, pacientes y sociedad en general, pues la catarata y otros problemas oculares desatendidos pueden ocasionar la pérdida de la visión repentina. Por otro lado, existen ciertas limitaciones presentes en el proyecto y un reto enorme recae en comprender las funcionalidades del software. En este caso, es necesario definir los objetivos del sistema a implementar y lo que se requiere analizar para emplear las técnicas pertinentes. Asimismo, armar un prototipo de la cabina requiere tiempo y acceso a dispositivos electrónicos de calidad, lo cual a veces resulta complicado debido a la sensibilidad de los componentes en las pruebas de proyecto.

Sin embargo, si se desea conocer su impacto potencial se sugiere que el procesamiento de imágenes sobre todo en Matlab presenta múltiples modalidades y puede ser empleado en medicina nuclear [2]. En específico en las radiografías computarizadas, para determinar la influencia de la radiación de rayos gamma en el cuerpo del paciente. Esto es posible gracias a los parámetros de contraste y representación de la energía que contiene el software. Además, con el desarrollo de un algoritmo basado en Deep Learning y con ayuda de fotografías médicas tomadas al fondo de ojo se puede examinar con mayor precisión el área de la retina [3], y así realizar un diagnóstico relacionado a problemas en la retina. Así, se ve que las técnicas de procesamiento facilitan una mejor visualización y acercamiento a las imágenes. Por añadidura, se pueden realizar transformaciones del algoritmo e implementar histogramas para dotar de eficiencia al método propuesto. De acuerdo con lo expuesto, el proyecto actual se convertirá en un sistema de prediagnóstico que estará al alcance de todo el público mediante una cabina con capacidad para una persona, donde se tomarán las fotos para su procesamiento en el software y posteriormente se enviarán al oftalmólogo mediante un enlace; lo cual simplificará el tiempo de atención para el diagnóstico. El proyecto pretende obtener un prediagnóstico de catarata y pterigión bajo parámetros establecidos en el procesamiento. Ante esto, la convolución neural en Matlab se convierte en una alternativa ideal para esclarecer las zonas infectadas o cubiertas con fluidos sanguíneos asociadas normalmente a patologías oculares [4]. Por último, se ha constatado la eficiencia de las técnicas de procesamiento [5]: primero, atribuyen el uso de filtrado y segmentación para enfocarse en las estructuras anatómicas de interés a profundidad; segundo, el proyecto Vismedic expuesto, está compuesto de programación en lenguaje C++, el lenguaje UML para el modelado y la herramienta CASE Visual Paradigm. Todo ello favoreció a la presentación de una interfaz gráfica que almacena las imágenes médicas y aplica el filtrado y segmentación mostrando una comparativa real con las imágenes ingresadas. De esta forma, se puede extender la aplicación sin modificar los parámetros y mejorar los beneficios que aporta. Bajo estas nociones se denota el enfoque del proyecto y la importancia de ciencias como ingeniería biomédica y mecatrónica, puesto que se relacionan entre sí con el fin de aportar el valor esperado

DESARROLLO

El proyecto se plantea la siguiente hipótesis: Si se implementa un procesador de imágenes considerando Matlab durante las tele consultas oftalmológicas para el prediagnóstico presuntivo de problemas oculares, entonces se podrá obtener beneficios con la automatización del proceso y se brindará tratamiento oportuno a los pacientes con ayuda del oftalmólogo y la tecnología actual.

Para demostrar y comprobar la hipótesis formulada, se operacionalizaron, determinando las variables e indicadores que a continuación se mencionan:

Variable independiente: Procesador de imágenes oculares con Matlab.

Esta variable tiene como indicador la anormalidad en la forma del iris, que se determinará por el diámetro del iris (di), es decir, el diámetro normal promedio del iris es de 13 mm según indica la patología.

El margen de error debe de ser menor a 5%; para su cálculo se efectuará la siguiente ecuación:

$$\%eror = \frac{|di-13|}{13} \times 100\% \qquad (1)$$

Variable dependiente: Prediagnóstico en oftalmología.

Esta variable tiene los siguientes indicadores:

Poco avance del problema oftalmológico, que se determina en base a la resta del diámetro del iris normal de 13mm y del encontrado en las pruebas, como se indica en la ecuación 2:

$$AC = 13 - di \qquad (2)$$

Dependiendo del resultado obtenido podremos determinar el nivel de daño encontrado en el iris. Es decir, si los valores de AC son iguales a 13 mm significa que no hay daño; si AC está entre 13mm y 9mm el daño es controlado; si AC es menor igual a 8mm el daño es alto y si AC es igual a 1mm, el daño será de carácter irreversible.

El otro indicador es la tasa de pacientes con problemas oculares (TPO), el cual se determinará con los pacientes con síntomas notorios (PSN) encontrados de una muestra de 9 personas, empleado la siguiente ecuación:

$$TPO = \frac{PSN}{TOTAL\ DE\ FAMILIARES} \times 100 \qquad (3)$$

El tercer indicador son los grados de apertura que tiene el pterigión, el cual indicará el grado de pterigión encontrado en las pruebas.

Al establecer los parámetros se procede a la ejecución del proyecto para lo cual se necesitaron los siguientes materiales:

Tabla 1: Materiales e instrumentos usados

MATERIALES E INSTRUMENTOS USADOS	
Unidad	Material
01	Arduino Uno R3
01	Sensor PIR HC-SR501
02	Resistor de 220Ω
02	Leds
01	Protoboard
07	Jumpers
01	Software Matlab

El desarrollo se realizará en etapas, comenzando por la comunicación del sensor con el software Matlab, luego por el procesamiento de las imágenes mediante códigos en Matlab y finalizando por el diseño de la cabina en el software de Autodesk Inventor.

Recepción de señal del sensor

El algoritmo inicia con un sensor cuando el paciente ingresa a la cabina. Este sensor se activa automáticamente mandando un pulso 'a' en un determinado tiempo al Arduino y tomando automáticamente una captura de imagen.

```
clc, clear all, close all
%%% Código de recepción del Arduino a MATLAB
s= serial('COM3','BaudRate',9600); %COM3 SE
EDITA DE ACUERDO CON SU SALIDA EN EL ARDUINO
fopen(s)
disp ('Inicio')
data = fscanf(s);
if data == 'a'
    disp ('Inicio proceso')
```

Figura 1: Código de la recepción de señal del sensor PIR.

Para la conexión física se conectan los 3 pines del sensor PIR al protoboard con los jumpers; para visualizar la recepción de movimiento por el infrarrojo se añade el led verde junto a la resistencia para evitar que éste se queme. Para el caso contrario, es decir, no detecte movimiento se conecta un led rojo junto a una resistencia de protección; luego se conecta la placa de Arduino para alimentar el protoboard y programar la señal del sensor. La simulación del funcionamiento del circuito se llevó a cabo en la nube de Tinkercad.

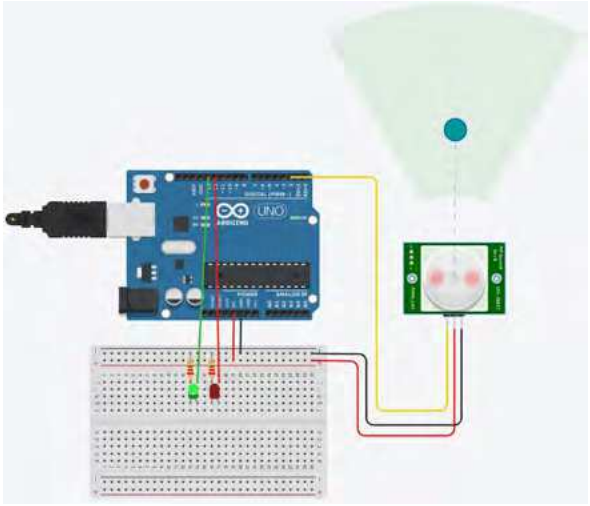


Figura 2: Conexión eléctrica en Tinkercad.

Toma de datos

El algoritmo continúa cuando se recibe la señal '1' del sensor y tomando la captura de la imagen mediante la aplicación de la IP WEBCAM.

```
IP='192.168.1.8:8080'; %Se edita de acuerdo con la red
que esté conectada
url=strcat('http://',IP,'/shot.jpg');
Icel=imread(url);
fh_real=image(Icel);
done=false;
n=1;
while ~done
    if ~strcmp(a,'1')
     Icel=imread(url);
     set(fh_real,'CData',Icel);
     drawnow;
     filename=sprintf('Img_%d.jpg',n);
     imwrite(Icel,filename,'jpg'); %Guardar imagen tomada
     n=n+1;
  I = imread(filename);
```

Figura 3: Código de la captura de imagen.

Procesamiento de imagen capturada

Una vez obtenida la imagen se procede a realizar el preprocesamiento para poder mejorar la imagen y tener una mejor vista del ojo.

```
%rreprocesamiento
  AInv = imcomplement(I); %Imagen invertida
  BInv = imreducehaze(AInv); %Reducción de neblina
  B = imcomplement(BInv); %Invertir resultado para obtener la imagen mejorada
  H = histeq(B); %Mejorar el contraste mediante la ecualización del histograma
  grayimage = rgb2gray(H); %Convertir la imagen RGB a escala de grises
```

Figura 4: Código del preprocesamiento de imagen.

Luego se procede a recortar ambos ojos, mediante el detector de rostros “Cascade”.

```
%Detecta los ojos de la imagen
EyeDetect=vision.CascadeObjectDetector('EyePairBig'); %Detección del par de ojos
bboxeyes = EyeDetect(grayImage);
I_eye=insertObjectAnnotation(grayImage,'rectangle',bboxeyes,'Ojos'); %Cuadro de detección de ojos etiquetado
figure (1)
imshow(I_eye); title('Detección de ojos');

%bboxeyes (x y width height) (Una incluyendo el par de ojos)
bboxeyeshalf=[bboxeyes(1,1),bboxeyes(1,2),bboxeyes(1,3)/2,bboxeyes(1,4)];
%Recortamos los dos ojos
leftEyeImage = imcrop(I_eye,bboxeyeshalf(1,:));
%leftEyeImage = imadjust(leftEyeImage); %Recorte del ojo izquierdo
rightEyeImage = imcrop(I_eye,[bboxeyes(1,1)+(bboxeyes(1,3)/3),bboxeyes(1,2),bboxeyes(1,3)/3,bboxeyes(1,4)]);
%rightEyeImage = imadjust(rightEyeImage); %Recorte del ojo derecho
```

Figura 5: Código de la detección del par de ojos.

Ya obtenido el par de ojos por separado, se binarizan las imágenes; el nivel de binarización se obtiene de los histogramas del par de ojos.

```
%Binarización iris
BWi=im2bw(leftEyeImage,0.25); %Binarización del ojo izquierdo con un valor umbral de 0.25
BWd=im2bw(rightEyeImage,0.37); %Binarización del ojo derecho con un valor umbral de 0.37
```

Figura 6: Código de la binarización del par de ojos.

Al realizar la binarización se encuentra que existen vacíos, por lo que se procede a rellenarlos con la función imfill, el cual iguala a 0 los píxeles dentro del área.

```
%Relleno de huecos
BWi=BWi==0;
BWd=BWd==0;
BWi=imfill(BWi,'holes');
BWd=imfill(BWd,'holes');
```

Figura 7: Código del relleno de huecos.

Para eliminar los rastros de pestañas y demás píxeles innecesarios, se identifica su área mediante las propiedades y define un rango entre 0 y 2300 para el iris.

```
%Eliminación de pixeles innecesarios en el iris

[L Ne]=bwlabel(BWi); %identifica los objetos presentes en lo imagen
%Calcular propiedades de los objetos presentes en la imagen
propied=regionprops(L,'basic'); %calculo de propiedades-iris izquierdo
hold on
%Buscar Áreas menores a 2300
a=find([propied.Area]<2300);
%Eliminar Áreas menores a 2300
for n=1:size(a,2)
    cajal=round(propied(a(n)).BoundingBox);
    BWi(cajal(2):cajal(2)+cajal(4),cajal(1):cajal(1)+cajal(3))=0;
end

[L1 Ne]=bwlabel(BWd); %identifica los objetos en el iris derecho
propied1=regionprops(L1,'basic');  %propiedades-iris derecho
a1=find([propied1.Area]<2300); %Busca Áreas menores a 2300
for n=1:size(a1,2)
    cajal=round(propied1(a1(n)).BoundingBox);
    BWd(cajal(2):cajal(2)+cajal(4),cajal(1):cajal(1)+cajal(3))=0;
end
```

Figura 8: Código para eliminación de pixeles innecesarios.

Por último, se aplica la detección de bordes con sobel para una mejor apreciación del iris y concluir si está siendo afectado o no. Esto se reflejará en la forma del iris, ya que si presenta una deformación indicará la presencia de pterigión o catarata. Para una mejor visualización de lo último expuesto es que se agregó un marco circular estableciendo rangos de detección por distancia del radio.

```
%Detección de bordes con sobel
%Bordes en el ojo izquierdo
BW1i = edge(BWi,'sobel');
%Bordes en el ojo derecho
BW1d = edge(BWd,'sobel');
```

Figura 9: Código para la detección de bordes.

Para una mejor apreciación de los contornos del iris se muestran los bordes detectados sobre la imagen original, además del círculo que representa a un iris normal, para ello se hacen unas modificaciones en el comando: mostrar imagen de Matlab.

```
%Circulo del iris
figure(2)
imshow(BWi);
d=imdistline; %Determinar el rango de radio de detección
figure(3)
[centers, radii]=imfindcircles(BWi,[24 70]); %Encuentra los circulos en ese rango
%Muestra los circulos encontrados el ojo izquierdo y los remarca en color rojo
subplot(1,2,1), imshow(leftEyeImage); viscircles(centers, radii,'Edgecolor','r');
title('Marco del ojo izquierdo')
hold on
%Marca el borde detectado sobre la imagen original para una mejor visualización
   B=bwboundaries(BW1i);%bw
 for k=1:length(B)
    boundary=B{k};
    plot(boundary(:,2),boundary(:,1),'y','LineWidth',2)
end
  hold off

[centers, radii]=imfindcircles(BWd,[24 70]); %Encuentra los circulos en ese rango
%Muestra los circulos encontrados el ojo derecho y los remarca en color rojo
subplot(1,2,2), imshow(rightEyeImage); viscircles(centers, radii,'Edgecolor','r');
title('Marco del ojo derecho')
hold on
%Marca el borde detectado sobre la imagen original en color amarillo
   B=bwboundaries(BW1d);%bw
 for k=1:length(B)
    boundary=B{k};
    plot(boundary(:,2),boundary(:,1),'y','LineWidth',2)
end
  hold off
end
    if strcmp(a,'q')
        done=true;
        return
    end
end
end
fclose(s); %Fin del bucle
```

Figura 10: Código para mostrar la imagen procesada final.

Resultado de las imágenes procesadas

Para cumplir con los objetivos del proyecto se tomaron 2 muestras que evidencien el correcto funcionamiento del código.

Figura 11: Resultado de la toma de un ojo con Pterigión.

En la Figura 11 se muestra la toma de una persona adulta, donde al procesar la imagen se evidencia la presencia de pterigión en ambos ojos; esto se concluye por la deformación del borde del iris ya que no es circular completa, puesto que presenta una deformación producida por la carnosidad que va del lagrimal del ojo al iris.

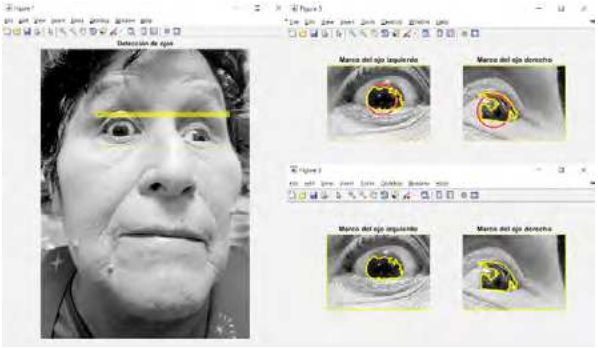


Figura 12: Resultado de la toma de un ojo con Catarata.

En la Figura 12 se observa a una mujer de tercera edad, que muestra signos de catarata cortical en el ojo izquierdo; esto se evidencia en la forma irregular en el iris y se determina que es de tipo cortical porque va del exterior del iris al interior.

Como resultado del proyecto, la tasa de pacientes con síntomas notorios testeados en nuestro entorno familiar es de 22,2%. Ello se obtiene de la ecuación 2. Dichos familiares presentan edad avanzada e indicios de catarata y pterigión debido a la deformación del iris en el procesamiento.

El porcentaje de error alcanzado en las pruebas es el esperado, debido a que no excede al 5%, pues se encuentra entre 2% y 3% lo cual señala la eficiencia del procesamiento.

De acuerdo con la ecuación 3 que determina el poco avance del problema oftalmológico, se comprobó que la muestra de 9 familiares se clasifica en la categoría de daño controlado.

Diseño de la cabina

Para una futura implementación del proyecto, se diseñó la cabina en Inventor con medidas milimétricas a escala real en la cual se efectuaría el prediagnóstico a distancia.

Para un futuro uso y pensando en la inclusión de todas las personas es que se agregó una rampa metálica, la cual está diseñada bajo las respectivas normas teniendo un ángulo inclinación menor a 12%.

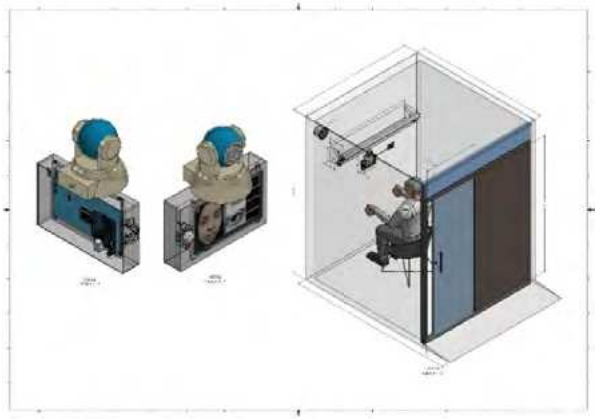


Figura 13: Diseño de la cabina.

CONCLUSIONES

Las técnicas de procesamiento permiten tener un panorama más claro y brindar un prediagnóstico, el cual requiere ser confirmado por un especialista. Sin embargo, durante este estado de emergencia se puede adoptar esta implementación.

La construcción de la cabina a escala real supone un gran reto, pero no tendrá alteraciones en los resultados. El uso del sensor PIR permite contener un ángulo de detección preciso y garantiza su eficacia en espacios cerrados como la cabina. El proyecto es accesible y supone ahorrar gastos, así como evitar el contacto directo entre paciente y médico.

Durante el proyecto existieron limitaciones; no obstante, con ayuda de la tecnología actual se logró desplazarlos y dar un giro continuo al proyecto. Asimismo, no es un fin del proyecto, pues con variadas herramientas como análisis de energía en capas, captación de señales y métodos complejos de

ingeniería biomédica y mecatrónica se puede dotar de versatilidad y adaptarlo a situaciones específicas teniendo en cuenta su objetivo.

AGRADECIMIENTOS

Se extiende un honorable agradecimiento a nuestros padres, por impulsarnos a diario y contribuir a cultivar nuestro aprendizaje. Asimismo, a nuestro docente debido a su vocación por la investigación en el área de ingeniería y su confianza en los conocimientos adquiridos por sus estudiantes.

REFERENCIAS

Anales de Congresos y Seminarios:

[1] Deperlıoğlu, Ö.; Köse, U. (2018). Diagnosis of Diabetic Retinopathy by Using Image Processing and Convolutional Neural Network. *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 1-5.

[2] Najeeb, A. ; Osman, A.; Al-Tijani A. Abdelrahman, A. (2018). Implementation of computed radiography in nuclear medicine imaging. *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 1-4.

[3] Karthiyayini, R.; Shenbagavadivu, N. (2020). Retinal Image Analysis for Ocular Disease Prediction Using Rule Mining Algorithms. *Interdiscip Sci*, 13(3), 451-462. Recuperado de:https://pubmed.ncbi.nlm.nih.gov/32514844/

[4] Liu, J.; Zhao, H. (2021). Application of convolution neural network in medical image processing. *Technol Health Care*, 29, 407-417. Recuperado de: https://pubmed.ncbi.nlm.nih.gov/33386836/

[5] Peña-Peñate A.; Silva L., Alcolea R. (2016). Módulo de filtrado y segmentación de imágenes médicas digitales para el proyecto Vismedic. *Revista Cubana de Ciencias Informáticas*, 10(1), 13-27.